

13/02/77

Titel: Verfahren und Vorrichtung zur Datenverarbeitung

Beschreibung

5

Die Erfindung betrifft das oberbegrifflich Beanspruchte und befaßt sich somit mit Verbesserungen bei multidimensionalen Feldern aus datenverarbeitenden Zellen zur Datenverarbeitung.

10

Es sind bereits multidimensionale Felder aus datenverarbeitenden Zellen bekannt. Zur Gattung dieser Bausteine zählen insbesondere systolische Arrays, neuronale Netze, Mehrprozessor Systeme, Prozessoren mit mehreren Rechenwerken und/oder logischen Zellen und/oder kommunikativen/peripheren Zellen (IO), Vernetzungs- und Netzwerkbausteine wie z.B. Crossbar-Schalter, ebenso wie bekannte Bausteine der Gattung FPGA, DPGA, Chameleon, XPUTER, etc.. Es sind insbesondere Bausteine bekannt, bei denen erste Zellen während der Laufzeit ohne Störung des Betriebes weiterer Zellen umkonfigurierbar sind, vgl. etwa die folgenden Schutzrechte und Anmeldungen desselben Anmelders: P 44 16 881.0-53, DE 197 81 412.3, DE 197 81 483.2, DE 196 54 846.2-53, DE 196 54 593.5-53, DE 197 04 044.6-53, DE 198 80 129.7, DE 198 61 088.2-53, DE 199 80 312.9, PCT/DE 00/01869, DE 100 36 627.9-33, DE 100 28 397.7, DE 101 10 530.4, DE 101 11 014.6, PCT/EP 00/10516, EP 01 102 674.7. Diese sind hiermit zu Offenbarungszwecken vollumfänglich eingegliedert.

30 Die dergestalt aufgebauten Bausteine weisen eine hohe Leistungsfähigkeit auf; allerdings ist deren Einsatz aufgrund der hohen Kosten oftmals prohibitiv. Wo Kosten bei hohen

Stückzahlen besonders relevant sind, ist es daher derzeit üblich, dedizierte Logikschaltungen in Form von ASICS und dergl. vorzusehen. Diese haben jedoch das Problem, besonders hohe Entwicklungskosten mit sich zu bringen, da sowohl der Entwurf der Schaltung als auch die Herstellung der Vielzahl von Masken teuer ist.

Die Aufgabe der vorliegenden Erfindung besteht darin, Neues für die gewerbliche Anwendung bereitzustellen.

10

Die Lösung dieser Aufgabe wird in unabhängiger Form beansprucht. Bevorzugte Ausführungsformen finden sich in den Unteransprüchen.

15 Gemäß einem ersten Aspekt der vorliegenden Erfindung wird somit vorgeschlagen, daß bei einer Datenverarbeitungsanordnung mit einem multidimensionalen Feld in Funktion und/oder Vernetzung konfigurierbarer Zellelemente und diesen zugeordneten Konfigurationsvorhalte-mitteln zum lokalen Konfigurations-
20 Vorhalten, vorgesehen ist, daß die Konfigurationsvorhaltemittel dazu ausgebildet sind, zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorzuhalten.

Es wird demnach vorgeschlagen, die Leistungsfähigkeit der multidimensionalen Prozessorfelder dadurch zu optimieren, daß
25 zunächst zwar eine Vielzahl von Zellen vorgesehen wird, die über se zu einer großen Vielzahl verschiedener Funktionen befähigt sind, aber dann aus dieser Vielzahl verschiedener Funktionen nur eine oder wenige Funktionen für jede Zelle
30 vorzusehen. Hierbei ergeben sich gegenüber dem dedizierten Schaltungsentwurf von Asics und dergl., bei denen exakt die jeweils für die benötigten Funktionen erforderlichen Schalt-

- kreise vorgesehen werden, gravierende Kostenvorteile, weil Rückgriff auf leicht programmierbare Einheiten oder durchgetestete Module genommen werden und so keine hohen Entwicklungs- und/oder Testkosten anfallen, und weil überdies
- 5 keine hohen Kosten für eine Vielzahl von Masken entfallen, die beim Entwurf dedizierter Asics sonst fällig sind. Der Entwurf kann über herkömmliche Entwurfsprogramme für logische Schaltungen erfolgen, in denen Module für die Zellen, Vernetzungsarchitekturelemente usw. bereitgestellt werden, oder
- 10 indem eine analoge, rekonfigurierbare Anordnung so konfiguriert wird, bis sie die gewünschten Ergebnisse liefert und dann die entsprechende Funktionalität in einer Anordnung fest vorgegeben wird.
- 15 Besonders bevorzugt ist es, wenn die die Funktion grobgranular konfigurierbar ist, d.h. wenn das Konfigurationsvorhaltemittel nur wenige Bit vorhalten muß, um eine jeweilige Funktion der Zelle zu bestimmen. Dies erleichtert es, eine Vielzahl sukzessive abzuarbeitender
- 20 Konfigurationen vorzuhalten, die aber jeweils oder zumindest zum Teil fest vorgegeben sind. Als Zellelemente können zumindest eines von ALUs, EALUs, RAM-Zellen, I/O-Zellen, Logiblöcken vorgesehen sein. Es kann auch die Vernetzung grobgranular konfigurierbar sein, d.h. es werden nur wenige
- 25 Bits zu setzen sein, um die Vernetzung vorzusehen. In einer alternativen Weise ist es möglich, die Vernetzung zumindest weitgehend fest vorzugeben und nur die jeweilige Funktion zu variieren. Dies ist dann bevorzugt, wenn der fertige Baustein etwa in seiner Funktion wie bei der Wave-Rekonfiguration je-
- 30 weils eine bestimmte einer vorgegebenen Anzahl von Funktionen ausführen soll, aber die Vernetzung selbst fest steht. Dazu kann in bestimm-ten Teilbereichen nur eine Nächste-Nachbar-

Verbindung vorgesehen werden (auf die parallel eingereichte Anmeldung des Anmelders betreffend die Erhöhung der Nächste-Nachbar-Dimensionalität bzw. -Konnektivität sei zu Offenbarungszwecken hingewiesen.), von denen einige der Nächste-

5 Nachbar-Verbindungen aktiviert und einige deaktiviert sind. In anderen Bereichen kann dagegen z.B. eine, erforderlichenfalls auch laufzeitrekonfigurierbar veränderliche Beschaltungsanordnung und/oder Busstruktur vorgesehen werden. Es sei darauf hingewiesen, daß abhängig von den Benutzeran-

10 forderungen eine Vielzahl unterschiedlicher Aufgaben mit einer bis auf die Konfigurationsvorgabe unveränderten Bausteinen vorgesehen werden kann, so daß sich Maskenkosten auf eine Vielzahl von Bausteinen verteilen und damit nicht mehr so stark ins Gewicht fallen.

15

Es ist bevorzugt, wenn jedem Zellelement ein eigenes Konfigurationsvorhaltemittel zugeordnet ist. Diese können die bei XPP-Architekturen vorgesehenen, von einem zentralen Konfigurationsspeicher zugreifbaren Konfigurationsregister ersetzen.

20 Es ist möglich, in den Konfigurationsvorhaltemittel eine Vielzahl von Konfigurationen vorzuhalten; dies erlaubt etwa die Rekonfiguration im Betrieb, ohne daß eine ebenfalls teure und Siliziumfläche erfordernde Konfigurationseinheit integriert sein muß. Die Auswahl der jeweils zu aktivierenden

25 Konfigurationen kann innerhalb des Feldes über Statustrigger, Datenoperationen, Sequenceranordnungen etc. erfolgen. Es ist dabei auch bevorzugt, wenn mehrere fest vorgegebene nichtflüchtige Konfigurationen im Konfigurationsvorhaltemittel vorgegeben sind. Alternativ sind flüchtige und nichtflüchtige

30 Konfigurationen einsetzbar. Es wird darauf hingewiesen, daß eine vollständige oder partielle Konfigurationsvorgabe vor der oder jeder eigentlichen Inbetriebnahme erfolgen kann. Da-

zu können auf geeignete Weise eingegebene Daten als Konfigurationen behandelt werden, die abzulegen sind. Da solche eine Vorab-Ablage von Rekonfigurationsdaten nicht betriebsstörungsfrei geschehen muß, eröffnen sich hier weitere

5 Möglichkeiten, die Architektur zu vereinfachen; auf das sog. Wormholerouting sei hingewiesen, das bei laufzeitrekonfigurierbaren Einheiten nicht funktioniert. Alternativ und/oder zusätzlich kann vorgesehen sein, daß bei einigen Zellen im Betrieb mit veränderlichen Konfigurationen versehbare Konfigurationsvorhaltemittel vorgesehen sind, d.h. ein Teil der
10 Zellen über einen Konfigurationsmanager oder auf andere Weise umkonfiguriert wird.

Die wechselnde der Vielzahl von vorgehaltenen und/oder vorbestimmten Konfigurationen, die jeweils zu verwenden ist,
15 kann insbesondere im Wege der Wave-Rekonfiguration oder des lokalen Sequencing bestimmt bzw. geändert werden.

Es ist möglich, die Konfigurationsvorhaltemittel auszubilden
20 als ROM, EPROM, EEPROM, Flash-Speicher, Fuse-, Antifuse-programmierbare Speichermittel und/oder in insbesondere in oberen Lagen einer Siliziumstruktur fest vorgesehene Speichermittel gewählt sind. Besonders bevorzugt sind Anordnungen, die leicht und einfach bei einer großen Stückzahl die Konfiguration vorsehen. Dies ist durch geeignete
25 Maskierung bei der Herstellugn auf den oberen Metalllagen erreichbar (z.B. Lage M4 und/oder M5) und/oder durch Fuse/Antifuse-Techniken. Letztere haben den Vorteil, daß bei Funktionsänderungen in einer laufenden Serie Änderungen
30 leichter implementierbar sind.

Mit der Anordnung kann ein Baustein festgelegter Funktion erhalten werden, indem ein multidimensionales Feld mit in Funktion und/oder Vernetzung konfigurierbaren Zellelementen und diesen zugeordneten Konfigurationsvorhaltemitteln zum lo-

5 kalen Konfigurations -Vorhalten vorgegeben wird, bestimmt wird, welche Konfigurationen in diesen vorzuhalten sind, und dann nichtflüchtige Konfigurationsvorhaltemittel so vorgesehen werden, daß sie zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorhalten. Es kann dabei von
10 einem zur laufzeitrekfigurierbaren multidimensionalen Feld ausgegangen werden, das ein höhere Funktionalität besitzt und es kann dann das Design um bestimmte Funktionen reduziert werden, bis ein Kern-Bauelement oder -elementblock mit vorgegebener Architektur erhalten wird, bei dem nur noch wenige
15 freie Konfigurationen zu bestimmen sind.

Dieser Aspekt der Erfindung wird beschrieben nur beispielhaft beschrieben mit Bezug auf die Zeichnung, in welcher zeigt

Fig A1 eine erfindungsgemäße Datenverarbeitungsanordnung

Fig A2 Details hierzu

Nach Fig. 1 umfaßt nun eine allgemein mit 1 bezeichnete Datenverarbeitungsanordnung 1 mit einem multidimensionalen Feld
25 in Funktion und/oder Vernetzung konfigurierbarer Zellelemente 2 und diesen zugeordneten Konfigurationsvorhaltemitteln 2a zum lokalen Konfigurations-Vorhalten, wobei die Konfigurationsvorhaltemittel 2a dazu ausgebildet sind, zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorzu-
30 halten.

Das multidimensionale Feld 1 umfaßt im vorliegenden Beispiele jeweils drei Reihen und Spalten PAEs, wie sie aus den Eingangs erwähnten und weiteren Veröffentlichungen des Anmelders per se bekannt sind. Diese Einheiten weisen grobgranular konfigurierbare ALUs 2b auf, an die über Multiplexer 2c Daten von einem Bussystem 2d eingehen und die beidseits mit Vorwärts-/Rückwärtsregistern 2e, 2f wie per se bekannt flankiert sind. Weiter speisen sie Ausgangsdaten über einen weiteren Multiplexer 2g auf ein Bussystem in der Reihe darunter auf.

Die Funktionsweise der Multiplexer 2g, 2c sowie jene der ALU 2b und der Register 2e, 2f ist per se bekannt und wird hier nicht detailliert erläutert. Die Konfiguration, die diese Einheiten haben, d.h. die Verbindung, die der Multiplexer jeweils aktiviert, bzw. die jeweilige Funktion der ALU, sind im Konfigurationsspeicher 2h abgelegt. Dabei können für Sequencing oder Wave-Rekonfiguration eine Vielzahl von unterschiedlichen Konfigurationen abgelegt sein, die auf Signale aus den Zellen oder auf externe Signale hin aktivierbar sind. Dabei muß nicht für alle Konfigurationen ein fester, unveränderlicher Speicher vorgesehen sein, sondern es kann auch in bestimmten Fällen ein (gegebenenfalls vergleichsweise kleiner) Speicher vorgesehen werden. Dies erlaubt demnach einen Zell- bzw. Speichermix.

Während in bisherigen Architekturen der Konfigurationsspeicher veränderlich war, und etwa von einer zentralen Konfigurationseinheit angesprochen wurde, ist im vorliegenden Fall der Konfigurationsspeicher 2h nichtflüchtig gebildet und sein Inhalt bei der Herstellung des die Elemente enthaltenden ICs festgelegt.

Dies geschieht wie folgt:

Zunächst wird festgelegt, welche Anzahl an Zellen und gegebenenfalls welche Zellen für die erwartete und mit der Datenverarbeitungsanordnung 1 abzuarbeitende Aufgabe erforderlich sind. Dann werden mit diesen die Funktion simuliert.

5 Das kann über Emulatoren geschehen oder es kann ein Feldlaufzeitrekonfigurierbaren Elemente mit zentraler Konfigurationseinheit zur Funktionsentwicklung bzw. zum Funktionstest herangezogen werden. Sobald die Funktionsentwicklung abgeschlossen ist und die erforderlichen Konfigurationen

10 festgelegt sind, wird ein Chip entworfen, der in seinem groben Aufbau einer Vielzahl anderer, gleichartiger Chips entspricht und sich lediglich hinsichtlich der nichtflüchtigen Konfigurationsspeicherinhalten von jenen unterscheidet.

15 Es wird dann festgelegt, ob die nichtflüchtigen Konfigurationsspeicherinhalte mit dedizierten Metall-Lagen festgelegt werden und/oder durch Brennen/Schmelzen bestimmter, zur Konfiguration vorgesehener Fuses/Antifuses oder auf andere Weise. Die Speicherinhalte werden dann während der Herstellung des Prozesses vorgesehen und der Chip ist für seine dedizierte Aufgabe ohne eine Vielzahl teurer Masken verwendbar. Dabei sind zB regionale Anpassungen möglich, zB, um
20 unterschiedliche Modems etc. zu implementieren.

25 In einem weiteren Aspekt der Erfindung befaßt sich diese mit der integrierten elektronischen Verarbeitung von Informationen, die in Form analoger Signale vorliegen. Dabei ist besonders hervorzuheben, daß etwa die analoge Verarbeitung, wie ersichtlich sein wird, auf fest vorgespeicherte Konfigurationen zurückgreifen kann, daß dafür aus unterschiedlichen
30 Konfigurationen auswählbar ist und daß etwa bestimmte Zellformen gleichfalls vorteilhaft sind. Für die integrierte

elektronische Verarbeitung von Informationen, die in Form analoger Signale vorliegen, existieren gegenwärtig mehrere Konzepte:

5 - Diskrete analoge, nicht programmierbare Bausteine, wie etwa Transistoren und Operationsverstärker;

- Analoge, programmierbare, integrierte Schaltkreise, genannt FPAAs (Field Programmable Analog Arrays), FPMAs (Field Programmable Mixed-Signal Arrays) oder FPADs (Field Programmable Analog Devices). FPAAs, FPMAs und FPADs bestehen ähnlich wie
10 die digitalen FPGAs (Field Programmable Gate Arrays) aus einzelnen, programmierbaren Zellen. Im Fall von FPAAs, FPMAs und FPADs ist das Kernstück einer solchen Zelle ein analoger
15 Operationsverstärker, dem eine bestimmte Funktion aus einem Satz möglicher Funktionen zugewiesen werden kann. Mögliche Funktionen sind zum Beispiel Addierer, Inverter, Gleichrichter und Filter erster Ordnung, mit denen ein analoges Signal bearbeitet werden kann. Die Zellen stehen untereinander durch
20 ein Bussystem in Verbindung und werden durch logische Elemente gesteuert;

- Anwendungsspezifische, nicht programmierbare integrierte Schaltkreise,
25 genannt ASICs (Application Specific Integrated Circuits);

- Programmierbare, voll digitale Prozessoren, genannt DSPs (Digital Signal Processors) oder CPUs (Central Processing Units), die der digitalen Verarbeitung analoger Signale nach
30 deren vorhergehender Analog-Digital-Wandlung dienen. Wenn nach der Verarbeitung wieder ein analoges Signal vorliegen

soll, muss nach der Verarbeitung eine Digital-Analog-Wandlung des Signals vorgenommen werden.

5 Probleme

Diskrete analoge Bausteine

Eine Schaltung mit diskreten Bauelementen kann aufgrund ihrer primären Flexibilität zwar optimal für eine bestimmte Aufgabe ausgelegt werden.

10

Die Aufgaben der Schaltung müssen allerdings zum Zeitpunkt des Schaltungsentwurfs genau bekannt sein, denn eine nachträgliche Anpassung der Schaltung an veränderte Anforderungen ist nicht oder nur mit erheblichem Aufwand möglich. Dies gilt insbesondere für die Programmierbarkeit und für Umkonfigurationen im Betrieb. Außerdem wird eine solche Schaltung bei komplexeren Aufgaben schnell umfangreich.

15

FPAAs, FPMAs, FPADs

20

Die durch FPAAs, FPMAs und FPADs gegebenen Möglichkeiten zur Verarbeitung analoger Signale orientieren sich am Vorbild klassischer analoger Signalverarbeitungsanlagen.

25

Sie sind für das zu verarbeitende Signal weitgehend transparent, das heißt, das zu verarbeitende Signal wird bis zu einer bestimmten, bausteinabhängigen Frequenz in Echtzeit bearbeitet.

30

Eine einfache Möglichkeit, analoge Werte zu speichern, existiert nicht, insbesondere nicht die Möglichkeit, den analogen Eingangswert und/oder den Ausgangswert jeder einzelnen Zelle zu speichern. Viele wichtige Operationen, wie etwa Schleifenberechnungen, und sämtliche Prozesse, bei denen mehrere Signale zeitlich koordiniert nacheinander verarbeitet

werden, werden durch Speicherung jedoch erst möglich. Eine einzelne FPAA-, FPMA- oder FPAD-Zelle kann zwar als Speicher nach Art einer Sample-and-Hold-Stufe konfiguriert werden. Sie kann dann jedoch keine zusätzliche Funktion mehr ausüben.

5

FPAAs, FPMAs und FPADs sind aufgrund ihrer ausschließlich analogen Signalverarbeitung funktionellen Einschränkungen unterworfen. Die Fähigkeiten der in FPAAs, FPMAs und FPADs implementierten digitalen Logik beschränken sich auf die Funktionen, die für die Umkonfiguration der Zellen notwendig sind. Die Funktion der Zellen, die diese während des Betriebs ausüben, wird durch die Logik im Stand der Technik nicht unterstützt, geschweige denn erweitert, etwa durch digitale Zählfunktionen oder logische Grundfunktionen wie beispielsweise NAND und NOR. Insbesondere gibt es keine jeweils zu einer einzigen Zelle gehörenden logischen Strukturen, die solche digitalen Zählfunktionen oder logischen Grundfunktionen durchführen können. Daß hier erfindungsgemäß Abhilfe geschaffen wird, sei im Vorgriff erwähnt.. Mit FPAAs, FPMAs und FPADs sind deshalb logische Funktionen wie zum Beispiel eingangssignalabhängige Entscheidungen, wenn überhaupt, nur in geringem Maße oder nur sehr aufwendig möglich.

Dasselbe gilt für die datenabhängige Rekonfiguration von FPAAs, FPMAs und FPADs, beispielsweise (aber nicht nur) nach Art einer IF-THEN-ELSE-Anweisung. Diese wird erfindungsgemäß ermöglicht. Soll eine FPAA-, FPMA- oder FPAD-Zelle aufgrund von Kriterien, die die zu verarbeitenden oder bereits verarbeiteten Analogsignale betreffen, rekonfiguriert werden, so muß das betroffene Analogsignal über eine temporäre oder sogar permanente Verbindung nach außen zu einer externen, nicht in dem FPAA, FPMA oder FPAD enthaltenen Struktur geführt

werden, welche über eine etwaige Rekonfiguration entscheiden und diese Rekonfiguration auslösen und durchführen muß. Es besteht für die Zelle keine Möglichkeit, abhängig von einem Analog- oder Digitalsignal selbständig, das heißt mit ihr eigenen Strukturen, über eine Rekonfiguration ihrer selbst zu entscheiden, diese Rekonfiguration zu veranlassen und die dazu notwendigen Daten von einer internen, auf dem Baustein enthaltenen, dafür geeigneten Struktur zu erhalten.

- 10 Will man das Ergebnis der Operation einer Zelle auf deren Eingang führen, zum Beispiel bei Schleifenoperationen, so kann das bei FPAAs, FPMAs und FPADs nur mittels des Busses geschehen; eine eigene Leitung zur Rückführung des Operationsergebnisses auf den Eingang der Zelle zur Entlastung des Busses ist bei FPAAs, FPMAs und FPADs nicht vorgesehen.

Die genannten Nachteile schließen es aus, mit FPAAs, FPMAs und/oder FPADs ein analoges Rechenwerk aufzubauen, das die Flexibilität und den Funktionsumfang heutiger digitaler Rechenwerke erreicht.

ASICs

- ASICs besitzen eine hohe primäre Flexibilität, da sie für eine spezielle Anwendung entwickelt werden. Sie eignen sich jedoch nur für diejenige Anwendung, für die sie entwickelt werden; rekonfigurierbar sind ASICs nur in demjenigen Rahmen, den die Anwendung vorgibt. ändert sich die Anwendung um ein Detail, welches bei der Entwicklung des ASICs nicht berücksichtigt wurde, so muß im Extremfall ein neuer ASIC entwickelt werden.

DSPs und CPUs

Unter allen Möglichkeiten zur Signalverarbeitung können DSPs und CPUs zwar am flexibelsten konfiguriert und rekonfiguriert werden, allerdings weder teilweise, noch während der Laufzeit.

5

Um Analogsignale in ein für DSPs oder CPUs geeignetes Format umzuwandeln, müssen die analogen Signale digital kodiert werden. Dies erfordert eine Analog-Digital-Wandlung, die bei höheren Anforderungen an die Präzision recht aufwendig und teuer werden und überdies noch die Bandbreite beschränken kann. Entsprechendes gilt für die Rücktransformation der digitalen, verarbeiteten Daten in Analogsignale. Um ausreichende Schnelligkeit zu erzielen, müssen die internen Bussysteme in DSPs und CPUs die einzelnen Bits eines digital kodierten Analogsignals parallel übertragen. Die erforderliche Breite des Datenbussystems wächst mit der geforderten Präzision der digitalen Kodierung des Signals. Im Gegensatz dazu genügt bei einer analogen Übertragung eine Leitung pro übertragenem Analogsignal.

20

DSPs und CPUs besitzen überdies keine zellartige Struktur, sondern sind in der klassischen von-Neumann-Architektur aufgebaut. Ihre Modularität ist deshalb nur gering.

25 Die heute existierenden analogen Rechenwerke erreichen bei weitem nicht den Funktionsumfang und die Konfigurierbarkeit heute existierender digitaler Rechenwerke.

Umgekehrt werden analoge Schaltungen zunehmend durch digitale Rechenwerke ersetzt, etwa im Fall der DSPs, wobei man die bei den DSPs genannten Nachteile in Kauf zu nehmen hat.

30 Die heute existierenden Methoden zur Verarbeitung analoger Signale haben zum Ziel, diese analogen Daten zu modifizieren.

Sind die dazu verwendeten Bausteine konfigurierbar, dann wird die Art und Weise, wie die analogen Signale zu modifizieren sind, ausschließlich durch digitale Logik eingestellt, das heißt, die Steuerung erfolgt ausschließlich durch digitale
 5 Signale. Es existieren weder Möglichkeiten, die Datenverarbeitungssteuerung unmittelbar durch analoge Signale vorzunehmen, noch Möglichkeiten, analoge Signale mit dem Funktionsumfang eines digitalen Rechenwerks zu bearbeiten.

- 10 Die Erfindung umfaßt somit auch ein programmierbares, zumindest teilweise analoges Rechenwerk (Reconfigurable Analog Processor, RAP) mit durch logische Elemente erweiterten Funktionen, in der Weise, daß der Funktionsumfang eines digitalen Rechenwerks verbunden wird mit der Möglichkeit zur schnellen,
 15 analogen Berechnung komplexer Funktionen (etwa der Logarithmusfunktion) und der Rekonfigurierbarkeit eines DFPs wie zb gemäß Offenlegungsschrift DE4416881A1.

- Ein RAP besteht aus Zellen, die in ihrer Funktion und Vernetzung frei konfigurierbar und während der Laufzeit rekonfigurierbar sind. Bei der Rekonfiguration einer einzelnen Zelle während der Laufzeit werden andere Zellen nicht in ihrer Arbeit beeinträchtigt. Eine Zelle ist unterteilt in eine Analogsektion und eine Logiksektion. Die Analogsektion
 20 dient der Verarbeitung analoger Daten auf der Basis von Operationsverstärkerschaltungen, wie sie von FPAA's, FPMAs und FPADs her bekannt sind. Die Logiksektion steuert die Funktionen der Analogsektion während der Laufzeit, bei der Anfangskonfiguration und bei der Umkonfiguration während der
 25 Laufzeit.
 30

Die Analogsektion kann aber auch analog gesteuert und konfiguriert werden. Die Datenverarbeitung findet wie bei FPAA's, FPMAs und FPADs in erster Linie analog statt; der Funktionsumfang wird jedoch durch besondere Strukturen mit jeweils
5 einer Logiksektion und verschiedenen Speichern in jeder Zelle dahingehend erweitert, daß in der Zelle eingangsdatenabhängige logische Operationen, Vergleiche, Schleifenoperationen und Zählvorgänge schnell und einfach durchgeführt werden können, so daß ein Funktionsumfang ähnlich dem eines voll digitalen
10 Rechenwerks erreicht wird.

Es besteht für jede RAP-Zelle zur Vereinfachung ihrer Rekonfiguration die Möglichkeit, abhängig von einem Analog- oder Digitalsignal selbständig, das heißt mit ihr eigenen, inter-
15 nen Strukturen, über eine Rekonfiguration ihrer selbst zu entscheiden, diese Rekonfiguration zu veranlassen und die dazu notwendigen Daten von einer dafür geeigneten Struktur zu erhalten.

20 Zwei unabhängige, rekonfigurierbare Bussysteme, eines für analoge Signale, das andere für digitale Signale, vernetzen die Zellen untereinander und mit der Außenwelt. Jedes analoge Signal benötigt zu seiner Übertragung nur eine analoge Busleitung. Bei einem digitalen Bus wächst die Zahl der
25 benötigten Leitungen bei paralleler Übertragung mit der geforderten Präzision der digitalen Codierung des analogen Signals stark an. Die notwendige Busbreite eines analogen Busses ist deshalb im Vergleich zu der eines digitalen Busses bei vergleichbarer Signalauflösung und Übertragungsrate ganz
30 wesentlich verringert. Es sei erwähnt, dass auf einem integrierten Schaltkreis Mischungen vorliegen können aus analogen und digitalen Schaltkreisen; dabei kann eine weitgehende

Trennung und/oder Übergangsbeschaltung z. B. in Form von DACs und/oder ACDs zwischen analogen und digitalen Elementen vorgesehen werden. Die digitalen Elemente können ihrerseits durch PAEs, RAM-PAEs usw. insbesondere mit geeignetem Aspekt-
5 verhalten gebildet sein.

Die Erfindung beschreibt in diesem Teilaspekt sonst u.a. ein analoges, umkonfigurierbares Rechenwerk (Reconfigurable Analog Processor, RAP) aus einzelnen funktionalen Zellen, die
10 durch ein geeignetes Bussystem untereinander und mit der Außenwelt verbunden sind. Die Funktion der Zellen ist konfigurierbar und kann während des Betriebs so rekonfigurierbar sein, daß dabei die Funktion anderer, nicht zu rekonfigurierender Zellen nicht beeinträchtigt wird. Eine
15 funktionale Zelle enthält eine Analogsektion und eine Logiksektion. Die Analogsektion dient der Verarbeitung analoger Daten auf der Basis von Operationsverstärkerschaltungen. Die Logiksektion steuert die Funktionen der Analogsektion während der Laufzeit, bei der Anfangskonfiguration und bei der Umkon-
20 figuration während der Laufzeit. Außerdem erweitert die Logiksektion die rein analogen Funktionen der Analogsektion durch die Bereitstellung von zb Logikfunktionen und/oder digitalen Zählfunktionen und/oder arithmetischen und/oder Speicherelementen. Jeder Zelle können ein oder mehrere analoge
25 Speicher zugeordnet sein, die analoge Größen wie beispielsweise Eingangs- oder Ausgangssignale speichern und zur weiteren Verarbeitung bereitstellen können. Außerdem gehören zu jeder Zelle ein oder mehrere digitale Register zur Speicherung von digitalen Daten, die für die Konfiguration
30 und den Betrieb der Zelle notwendig sind.

Für jede Zelle besteht die Möglichkeit, abhängig von einem Analog- oder Digitalsignal selbständig, das heißt mit ihr eigenen internen Strukturen, über eine Rekonfiguration ihrer selbst gegebenenfalls zu Gruppen zusammengefaßter Zellen oder anderer Zellen zu entscheiden, diese Rekonfiguration zu veranlassen und die dazu notwendigen Daten von einer dafür geeigneten Struktur, welche sich auf dem Baustein befinden kann, zu erhalten. Es existiert weiterhin die Möglichkeit, das analoge Ergebnis der Operation einer Zelle ohne Zugriff auf ein Bussystem auf den analogen Dateneingang der Zelle zurückzuführen.

In diesem Abschnitt werden Begriffe verwendet, deren Bedeutung von der allgemein gebräuchlichen in manchen Punkten abweichen kann. Zum besseren Verständnis folgen die Begriffsdefinitionen, wie sie in diesem Abschnitt verwendet werden.

Ein **Signal** soll hier definiert sein als eine Größe, beispielsweise eine Spannung $U_0(t)$, die zu einem bestimmten Zeitpunkt an einem bestimmten Punkt einer Schaltung herrscht. Ein solcher Punkt kann beispielsweise ein Ausgang, ein Eingang oder eine Busleitung sein. Die Spannung $U_0(t)$ kann entweder auf Masse (GND) oder auf eine zweite Spannung $U_1(t)$ bezogen sein. Das Signal kann zeitlich konstant oder zeitlich veränderlich sein.

Information soll hier definiert sein als Anzahl der möglichen, unterscheidbaren Zustände, die ein Signal annehmen kann.

Als **digitales Signal** oder **Digitalsignal** soll hier ein Signal dann bezeichnet werden, wenn es nur zwei Zustände, beispiels-

weise 0 oder 1, annehmen kann, also nur zwei Informationen im Sinne der hier verwendeten Definition der Information beinhaltet.

- 5 Als **analoges Signal** oder **Analogsignal** soll hier ein Signal dann bezeichnet werden, wenn es mindestens drei und höchstens abzählbar unendlich viele Zustände annehmen kann, also mehr als zwei Informationen im Sinne der hier verwendeten Definition der Information beinhaltet. Das bedeutet insbesondere,
- 10 daß mittels analogen Signalen immer mehr Informationen gleichzeitig über eine Leitung übertragen werden können als mit digitalen Signalen.

- Im folgenden wird der Aufbau einer erfindungsgemäßen funktio-
- 15 nalen Zelle und der Aufbau des zugehörigen, die Zellen vernetzenden Bussystems beschrieben.

Die Zelle

- 20 Eine Zelle stellt die kleinste vollständige, selbständige funktionale Einheit eines RAPs dar. Dabei sind zwei verschiedene Typen von Zellen möglich -- die einfache Zelle und die erweiterte Zelle. Beide Zelltypen können auf einem RAP zum Einsatz kommen. Sie unterscheiden sich im Funktionsumfang.
- 25 Beiden Zelltypen gemeinsam ist die Unterteilung ihrer Struktur in eine Analogsektion und eine Logiksektion.

- Einige oder alle Zellen können einen Taktvervielfacher zur Erzeugung eines lokalen, auf die Zelle beschränkten höheren
- 30 Taktes beinhalten, der beispielsweise die Zählfunktionen der Logiksektion der Zelle unterstützt. Denkbar ist auch, daß einige oder alle Zellen Strukturen zur Erzeugung eines

zellinternen oder lokal begrenzten Zelltakts beinhalten können, dessen Frequenz unabhängig von der Frequenz eines etwaigen Bustakts konfiguriert werden kann. Der Zelltakt kann aktivierbar und deaktivierbar sein.

5

Die einfache Zelle (SCELL)

Die Elemente der einfachen Zelle (SCELL) gliedern sich in
 10 zwei Gruppen, genannt Analogsektion und Logiksektion. Die Analogsektion dient der analogen Datenverarbeitung der analogen Eingangssignale einer Zelle, kann aber auch analoge Signale erzeugen, wie beispielsweise (aber nicht nur) ein Rechtecksignal oder ein Dreiecksignal. Die Logiksektion
 15 stellt zusätzliche nicht-analoge Funktionen zur Verfügung, insbesondere zB eingangsdatenabhängige logische Operationen, Vergleiche und Zählvorgänge, Speicher und/oder arithmetische Operationen und steuert darüberhinaus die Tätigkeit der gesamten SCELL. Ein Element der Logiksektion ist die
 20 Steuerlogik (CL). Sie steuert die Funktionen der Analogsektion und verwaltet Signale zur Rekonfiguration der Zelle, die über die Bussysteme erhalten oder abgeschickt werden.

Die Analog-Eingangsstufe der SCELL ist ein Multiplexer (MUX0)
 25 nach Stand der Technik für Analogsignale. Das zu verarbeitende analoge Signal wird von einem analogen Datenbussystem (ABUS) auf die Eingänge von MUX0 geführt. MUX0, gesteuert von der CL, selektiert das von der SCELL zu verarbeitende Analogsignal und schaltet es zu der analogen
 30 Verarbeitungseinheit (APU, Analog Processing Unit) durch. Die APU ist eine konfigurierbare Einheit nach Stand der Technik. Sie enthält eine oder mehrere Operationsverstärkerschal-

tung/en, deren Funktion aus einem Satz möglicher Funktionen ausgewählt werden kann. Die Auswahl der Funktion geschieht mittels eines digitalen Signals von der CL.

5 Funktionen der APU können beispielsweise (aber nicht nur) sein:

- Addition einer programmierbaren Größe zum analogen Eingangssignal der APU
- 10 - Subtraktion einer programmierbaren Größe vom analogen Eingangssignal der APU
- Multiplikation des analogen Eingangssignals der APU mit einer programmierbaren Größe
- Division des analogen Eingangssignals der APU durch eine
- 15 programmierbare Größe, Division einer programmierbaren Größe durch das analoge Eingangssignal der APU
- Logarithmierung des analogen Eingangssignals der APU
- Antilogarithmierung des analogen Eingangssignals der APU
- Invertierung des analogen Eingangssignals der APU
- 20 - Keine Veränderung des analogen Eingangssignals der APU
- Filterfunktionen, beispielsweise Hochpässe, Tiefpässe, Bandpässe und Notchfilter
- Signalerzeugung, beispielsweise Rechtecksignale, Dreieckssignale und Sinussignale mit programmierbaren Zeitkonstanten
- 25 - Potenzierung
- -Speicherung

Das zu verarbeitende analoge Signal wird entsprechend der durch die CL programmierten Funktion in der APU verändert oder (in der Funktion eines Spannungsfolgers) nicht verändert, oder die APU dient der Erzeugung eines neuen analogen

30 Signals. Denkbar ist insbesondere die Erzeugung eines Signals, das eine Rekonfigurationsaufforderung darstellt, und

in dem die notwendigen Rekonfigurationsparameter in analoger Form kodiert sind. Der analoge Ausgang der APU ist an eine Speicherstufe (BIPS) angeschlossen. Die BIPS kann sich in einem von mehreren durch die CL programmierbaren Zuständen befinden, beispielsweise in einem der folgenden.

10 BUFNONINV: Das Ausgangssignal der BIPS hat denjenigen Wert, der an ihrem Eingang lag, als die BIPS ein BUFFER-Signal von der CL erhielt. Der Ausgangswert wird konstant gehalten, solange das BUFFER-Signal anliegt.

15 BUFINV: Das Ausgangssignal der BIPS hat denjenigen invertierten Wert, der an ihrem Eingang lag, als die BIPS ein BUFFER-Signal von der CL erhielt. Der Ausgangswert wird konstant gehalten, solange das BUFFER-Signal anliegt.

INVERT: Das Eingangssignal der BIPS wird invertiert.

20 PASS: Die BIPS schleift das Eingangssignal unverändert durch.

3STATE: Der Ausgang der BIPS nimmt einen hochohmigen Zustand ein.

25 Der Ausgang der BIPS ist mit dem Eingang eines analogen Demultiplexers (DeMUX) verbunden, dessen Ausgänge mit den Busleitungen des ABUS verbunden sind. Über die CL wird gesteuert, auf welchen Ausgang des DeMUX das verarbeitete analoge Signal geführt wird.

30 Als zusätzliches Element der Logiksektion einer SCELL zur Erweiterung des Funktionsumfanges der SCELL existiert die

LOGUNIT. Sie ist in der Lage, beispielsweise folgende Funktionen durchzuführen:

5 - digitale Zähler, die von der CL und/oder der APU gesetzt, getriggert, abgefragt rückgesetzt und angehalten werden können; diese können als grobgranulare Logikelemente gebildet sein; andere grobgranulare Logik- und/oder Funktionselemente wie arithmetische, insbesondere ALUartige und/oder speichernde Elemente sind gleichfalls implementierbar.

10

15 - logische Grundfunktionen, wie NAND, NOR, AND, OR, XOR, INVERT, BUFFER, die aus der CL und/oder APU stammende Informationen logisch miteinander verknüpfen können. Hier handelt es sich also um feingranulare Logikelemente. Solche Informationen können abhängig vom Status der CL und/oder der APU sein, und/oder von zu verarbeitenden Signalen. Insbesondere können solche Informationen Kriterien sein, die auch zur Bildung eines RECONREQ-Signals (Rekonfiguration-Request) führen.

20

Die erweiterte Zelle (ECELL)

25 Die erweiterte Zelle (ECELL) enthält in einer bevorzugten Ausführungsform eine vollständige, voll funktionale SCELL, die um zusätzliche Elemente und Funktionen erweitert wurde, um insbesondere (aber nicht nur) Schleifenoperationen ohne Zugriff auf das Bussystem durchführen zu können.

30 Die analoge Eingangsstufe (MUX0) ist um einen zweiten, gleichwertigen, auf den ABUS zugreifenden analogen Multiplexer (MUX1) erweitert. Mit MUX0 und MUX1 ist es möglich, statt (wie bei einer SCELL) nur einem Eingangssignal zwei Eingangs-

signale zur anschließenden Verarbeitung in der Zelle freizugeben. Außer den Busanschlüssen besitzen MUX0 und MUX1 jeweils zusätzlich noch einen Eingang, der auf Masse gelegt ist und einen Eingang, auf den das Ergebnissignal vom Ausgang der BIPS der ECELL zurückgeführt wird. Der Ausgang von MUX0 führt das von MUX0 zur Verarbeitung selektierte Analogsignal, welches ausdrücklich auch der konstante Massepegel oder das Ergebnissignal vom Ausgang der BIPS der ECELL sei kann. Der Ausgang von MUX1 führt das von MUX1 zur Verarbeitung selektierte Analogsignal, welches ebenfalls auch der konstante Massepegel oder das Ergebnissignal vom Ausgang der BIPS der ECELL sei kann.

Die Ausgangssignale von MUX0 und MUX1 werden auf die folgenden, programmierbaren Speicherstufen (BUFF0, BUFF1) geführt. BUFF0 erhält das Ausgangssignal von MUX0, BUFF1 erhält das Ausgangssignal von MUX1. BUFF0 und BUFF1 sind durch die CL konfigurierbare Einheiten, deren Funktion aus einem Satz möglicher Funktionen ausgewählt werden kann. Mögliche Funktionen von BUFF0 und BUFF1 sind beispielsweise

BUFNONINV: Der Wert des Ausgangssignals von BUFF0 bzw. BUFF1 ist gleich demjenigen analogen Eingangssignal, das anlag, als BUFF0 bzw. BUFF1 ein BUFFER-Signal von der CL erhielt. Der Ausgangswert wird konstant gehalten, solange das BUFFER-Signal anliegt.

BUFINV: Der Wert des Ausgangssignals von BUFF0 bzw. BUFF1 ist gleich demjenigen analogen Eingangssignal, das anlag, als BUFF0 bzw. BUFF1 ein BUFFER-Signal von der CL erhielt. Der Ausgangswert wird konstant gehalten, solange das BUFFER-Signal anliegt.

INVERT: Das aktuelle analoge Eingangssignal von BUFF0 bzw. BUFF1 wird invertiert.

- 5 PASS: BUFF0 bzw. BUFF1 schleift das aktuelle Eingangssignal unverändert durch.

Das Ausgangssignal von BUFF0 und das Ausgangssignal von BUFF1 werden auf jeweils einen analogen Eingang der erweiterten
10 analogen Verarbeitungseinheit XAPU der ECELL geführt. Alle Funktionen der APU einer SCELL sind in der XAPU einer ECELL enthalten.

Im Gegensatz zur APU der SCELL besitzt die XAPU zwei analoge Eingänge, so daß in der XAPU Operationen mit zwei analogen,
15 zeitlich konstanten oder zeitlich veränderlichen Signalen möglich sind, insbesondere die Addition, Subtraktion, Multiplikation und Division zweier solcher Signale. Es ist damit denkbar, die XAPU mittels eines analogen, zeitlich konstanten oder zeitlich veränderlichen Steuersignals zu programmieren,
20 indem bestimmten Werten des Steuersignals bestimmte Funktionen zugewiesen werden. Darüberhinaus ist es denkbar, mit einem analogen Steuersignal der APU einen zur Ausübung einer Funktion notwendigen Parameter zu übermitteln. Wenn beispielsweise $f(t)$ ein analoges zeitlich veränderliches
25 (Spannungs-)Signal ist, welches mit einem zeitlich veränderlichen (Spannungs-) Signal $g(t)$ multipliziert werden soll, kann die XAPU dann als Multiplikator nach Art eines spannungsgesteuerten Verstärkers (Voltage Controlled Amplifier, VCA) nach Stand der Technik programmiert werden, wobei $f(t)$
30 an einem analogen Eingang der XAPU liegt, während $g(t)$ am anderen analogen Eingang der XAPU liegt und das besagte Steuersignal darstellt.

Das Ausgangssignal der XAPU wird auf den Eingang der BIPS geführt. Die BIPS der ECELL und die BIPS der SCELL können gleich sein. Das Ausgangssignal der BIPS wird auf den Eingang des DeMUX geführt. Der DeMUX der ECELL und der DeMUX der SCELL können gleich sein. Außerdem wird das Ausgangssignal der BIPS über eine separate Leitung auf einen Eingang von MUX0 sowie auf einen Eingang von MUX1 geführt.

Die Logiksektion kann ein Element zur Taktvervielfachung enthalten, welches den Takt des DBUS vervielfacht, und das programmierbar sein kann. Damit kann die ECELL intern mit einem Vielfachen des DBUS-Taktes operieren.

15 Rekonfiguration einer Zelle (cellreconfig)

Das RECONREQ-Signal

Die Analogsektion und die Logiksektion der Zelle sind bevorzugt in der Weise strukturiert und verbunden, daß die Zelle bei Eintreten bestimmter Kriterien ein Signal, das RECONREQ-Signal, erzeugen kann, mit welchem sie ihre eigene Rekonfiguration oder die Rekonfiguration einer anderen oder mehrerer anderer Zellen veranlassen kann. Das RECONREQ-Signal kann digital sein und über ein separates digitales Bussystem

weitergeleitet werden. Es kann aber auch analog sein und über ein separates analoges Bussystem weitergeleitet werden.

Mit einem analogen RECONREQ-Signal ist es möglich, neben den RECONREQ-Informationen noch zusätzliche Informationen, zum Beispiel die Adresse der zu rekonfigurierenden Zelle oder der zu rekonfigurierenden Zellen, gleichzeitig auf nur einer Busleitung zu übertragen.

Kriterien, die ein RECONREQ-Signal auslösen, können zum Beispiel (aber nicht nur) sein:

5 - Ein bestimmter Signalpegel, der von in der Zelle auftretenden Analogsignalen (zu denen auch die analogen Eingangs- und Ausgangssignale zählen) erreicht, überschritten oder unterschritten wird.

10 - Eine bestimmte Signaldifferenz, die zwischen in der Zelle auftretenden Analogsignalen (zu denen auch die analogen Eingangs- und Ausgangssignale zählen), erreicht, überschritten oder unterschritten wird.

15 - Eine bestimmte zeitliche Änderung eines Signalpegels, die von in der Zelle auftretenden Analogsignalen (zu denen auch die analogen Eingangs- und Ausgangssignale zählen) erreicht, überschritten oder unterschritten wird.

20 - Das Verstreichen einer bestimmten Zeitspanne.

- Das Auftreten eines bestimmten digitalen Signals oder einer bestimmten Kombination digitaler Signale in der Zelle oder an den digitalen Eingängen und/oder Ausgängen der Zelle.

25 Die in der obigen Auflistung genannten Signale können ausdrücklich auch von anderen Zellen oder weiteren Elementen des RAPs stammen. Außerdem können durch logische Verknüpfung (AND, OR, NAND, NOR, XOR usw.) der genannten Kriterien weitere Kriterien gebildet werden. Die Logiksektion der ECELL
30 enthält zur logischen Verknüpfung von Kriterien geeignete Strukturen, zB für Ergebnisvergleich, Flags einer ALU wie Übertrag einer arithmetischen Einheit (carry etc)

Die Kriterien zur Bildung eines RECONREQ-Signals werden in der CL der Zelle ausgewertet. Die CL der Zelle generiert aus diesen Kriterien ein digitales Wort (RECONREQ-Wort) mit den
5 nötigen RECONREQ-Informationen.

Dieses RECONREQ-Wort kann in digitaler oder analoger Form von der Zelle weitergegeben werden. Dafür stehen eigene Bussysteme (RECONREQ-Bus), ein digitaler Bus und ein analoger Bus, zur Verfügung.

10

Soll das RECONREQ-Wort in analoger Form weitergegeben werden, so wird das digitale RECONREQ-Wort in einem Digital-Analog-Umsetzer (DAC) in analoge Form gebracht. Jede Zelle kann zu diesem Zweck einen solchen DAC besitzen.

15

Die Daten, die zur Rekonfigurierung der Zelle notwendig sind, stellen eine dafür geeignete Struktur zur Verfügung. Diese Struktur kann beispielsweise eine Ladelogik und eine Switching-Tabelle sein, wie sie in Patentanmeldung DE196 54

20 846.2 beschrieben sind.

Die Ladelogik

Die Ladelogik (LL) ist eine Struktur, die nach einem
25 RECONREQ-Signal die Rekonfiguration der betreffenden Zelle oder der betreffenden Zellen durchführt. Mehrere Zellen stehen mit jeweils einer einzigen LL über den RECONREQ-Bus in Verbindung. Diese Zellen bilden mit der zugehörigen LL einen Cluster. Jede Zelle eines Clusters kann ein RECONREQ-Signal
30 an ihre LL absetzen und so jede Zelle desselben Clusters zur Rekonfiguration auffordern. Andere Möglichkeiten, eine Rekonfiguration anderer Zellen auszulösen, bestehen gleichfalls.

Verwiesen wird auf die o.g. Schriften und weitere Schriften der vorliegenden Anmelderin. Ein Baustein kann mehrere Cluster enthalten. Die LLs dieser Cluster stehen untereinander über ein Bussystem in Verbindung und können somit Informationen austauschen. Solche Informationen können insbesondere Adressen von umzukonfigurierenden Zellen sein. Dadurch ist es jeder beliebigen Zelle des RAPs möglich, jede beliebige Zelle des RAPs zur Rekonfiguration aufzufordern.

- 10 Die LL kann entsprechend PACT_SWT (vergl. zitierte Patentanmeldung) aufgebaut sein und kann damit digitale RECONREQ-Worte direkt verarbeiten. Die LL benötigt für die Verarbeitung eines analogen RECONREQ-Wortes jedoch analoge Vorstufen, nämlich eine analoge Selektierstufe (ASELSTAGE) und eine Analog-Digital-Wandlerstufe (ADC). Die Aufgabe der ASELSTAGE ist es, zu prüfen, ob und an welchem analogen RECONREQ-Bus ein RECONREQ-Signal anliegt. Ist ein RECONREQ-Signal auf einem analogen RECONREQ-Bus vorhanden, so wird dieser Bus von der ASELSTAGE selektiert und zur weiteren Verarbeitung auf den ADC geschaltet, welcher das analoge RECONREQ-Wort in ein digitales RECONREQ-Wort zurückwandelt, das von der LL verarbeitet werden kann.

- Die ASELSTAGE kann auf verschiedene Arten realisiert werden. Eine Möglichkeit ist die Verwendung eines Multiplexers, eine andere die Verwendung eines Arbiters.

- ASELSTAGE als Multiplexer.** Die analogen RECONREQ-Busse der von der LL überwachten Zellen liegen an den Eingängen eines getakteten Analogmultiplexers nach Stand der Technik. Bei jedem Takt wird der Multiplexer um einen Eingang weitergeschaltet, so daß bei jedem Takt ein anderer Bus am

Ausgang des Multiplexers liegt. Ein Komparator überwacht den Ausgang des Multiplexers. Wenn kein analoges RECONREQ-Signal am Ausgang des Multiplexers liegt, hat der Ausgang des Multiplexers einen bestimmten Pegel, beispielsweise 0 Volt. Liegt ein RECONREQ-Signal an, liegt ein anderer Pegel am Ausgang des Multiplexers, was den Komparator dazu veranlaßt, das RECONREQ-Signal auf den nachfolgenden ADC zu schalten. Alternativ und/oder zusätzlich können mehrere Komparatoren vorgesehen sein, die das Signal mit unterschiedlichen Signalpegeln vergleichen und so unmittelbar eine Auswertung bewirken. Dies bietet sich insbesondere an, wenn nur wenige Signalstufen zu unterscheiden sind.

ASELSTAGE als Arbiter. Die analogen RECONREQ-Busse der Zellen eines Clusters werden zunächst auf die Eingänge eines analogen Multiplexers (AMUX) geführt. Liegt an einem der analogen RECONREQ-Busse ein RECONREQ-Signal an, so wird dieser Bus durch den AMUX selektiert und das anliegende RECONREQ-Wort auf den Ausgang des AMUX geschaltet.

Bussysteme

Ein RAP enthält bevorzugt zumindest zwei voneinander unabhängige, flexible Bussysteme zur Vernetzung der einzelnen Zellen und zur Verbindung des RAPs mit der Außenwelt. Die bevorzugten Bussysteme können konfiguriert und während der Laufzeit rekonfiguriert werden, ohne daß die Tätigkeit des RAP unterbrochen werden muß. Die Bussysteme können mit Eigenschaften ausgestattet sein, wie sie in Patentanmeldung DE 197 04 742.4 beschrieben sind. Unterschieden wird hier das analoge Bussystem und das digitale Bussystem.

Das analoge Bussystem (ABUS)

Das analoge Bussystem ABUS dient der Übermittlung der zu bearbeitenden, bereits bearbeiteten oder neu erzeugten analogen Daten und analogen Signale von außen an die Zellen und/oder zwischen den Zellen. Insbesondere ist es mit dem ABUS möglich, Zellen zu kaskadieren, um auf diese Weise ein analoges Signal in mehreren aufeinanderfolgenden Operationen zu bearbeiten, wobei eine Operation von jeweils einer Zelle durchgeführt wird.

Der ABUS kann mit jeder seiner Leitungen mehrere, insbesondere mehr als zwei Informationen gleichzeitig übertragen, zum Beispiel 256 Informationen. Der ABUS kann mit einer festen oder variablen Frequenz getaktet sein oder asynchron, das heißt nicht getaktet, sein. Die Implementierung des ABUS kann in einer Art und Weise erfolgen, wie sie in Patentanmeldung DE 197 04 742.4 beschrieben ist.

Das digitale Bussystem (DBUS)

Neben dem ABUS existiert auf dem RAP ein zweites Bussystem, genannt DBUS.

Der DBUS ist getaktet und dient der Distribution digitaler Daten, beispielsweise Konfigurationsdaten und Statusdaten, zwischen den Zellen. Die Logiksektion jeder Zelle ist an den DBUS angeschlossen. Die Implementierung des DBUS kann in einer Art und Weise erfolgen, wie sie in Patentanmeldung DE 197 04 742.4 beschrieben ist.

Dieser Aspekt der Erfindung wird mit Bezug auf die Zeichnung nachfolgend beispielhaft erläutert, wobei dargestellt ist durch

Figur B1 zeigt den Aufbau einer einfachen Zelle
Figur B2 zeigt den Aufbau einer erweiterten Zelle

Figur B3 zeigt eine mögliche Art der Realisierung von BUFF0 bzw. BUFF1

Figur B4 zeigt, wie beispielsweise der Ausdruck $f(t)^g(t)$ berechnet werden kann.

5

Figur 1 zeigt den Aufbau einer einfachen Zelle (SCELL). Sie besteht aus der Digitalsektion (0101) und der Analogsektion (0102). Zentrales Element der Logiksektion ist die Steuerlogik CL (0110), die über den DBUS (0130) mit anderen Zellen, zusätzlichen Strukturen wie beispielsweise einer Ladelogik und/oder einer Switching-Tabelle, wie sie in Patentanmeldung DE 196 54 846.2 beschrieben sind, und/oder der Außenwelt kommunizieren kann.

15

Der Multiplexer MUX0 (0121) ist an den ABUS (0131) angeschlossen. Sofern ein Analogsignal von der SCCELL zu verarbeiten ist, selektiert MUX0 (0121), über die Leitungen (0141) von der Steuerlogik CL (0101) oder einer anderen geeigneten Struktur gesteuert, diejenige Leitung des ABUS (0131), auf der das zu verarbeitende Analogsignal anliegt.

20

Der Ausgang von MUX0 (0121) ist über die Leitung 0146 mit der analogen Verarbeitungseinheit APU (0120) verbunden. In ihr wird das von MUX0 selektierte Signal verarbeitet, sofern ein Signal selektiert wurde, oder die APU generiert ein Signal, welches ein RECONREQ-Signal sein kann, oder die APU verharrt in einem vordefinierten Ruhezustand. Das Verhalten der APU wird von der CL (0101) über die Leitungen 0143 gesteuert. Diese Leitungen (0143) können bidirektional

25

30

ausgeführt sein, so daß die APU in der Lage ist, abhängig von bestimmten Ereignissen und Kriterien Signale an die CL (0101) zu schicken. Die Kriterien können solche sein, die bei-

spielsweise auch zur Erzeugung eines RECONREQ-Signals führen. Ein erzeugtes Signal kann insbesondere ein RECONREQ-Signal, wie in Abschnitt *cellreconfig* beschrieben, sein. Das von der APU verarbeitete oder erzeugte Signal gelangt über die Leitung 0149 auf eine Speicherstufe BIPS (0124), deren Funktion von der CL (0101) gesteuert wird. Dabei stehen die in Abschnitt *scell* beschriebenen Funktionen BUFNONINV, BUFINV, INVERT, PASS, 3STATE zur Verfügung. Am Ausgang der BIPS wird das Analogsignal von einem Demultiplexer DeMUX (0125) übernommen, der es, gesteuert von der CL über Leitung 0145 oder einer anderen geeigneten Struktur, auf den ABUS 0131 schaltet.

Die Logiksektion (0101) der SCELL besteht aus der CL (0110) und der LOGUNIT (0111), die über die Leitung 0140 miteinander in Verbindung stehen.

Figur 2 zeigt den Aufbau einer erweiterten Zelle (ECELL). Sie ist funktionell unterteilt in eine Analogsektion (0202) und eine Logiksektion (0201). Die analogen Multiplexer MUX0 (0221) und MUX1 (0222) selektieren, gesteuert von der CL (0210) der ECELL, die beiden Analogsignale, die von der ECELL verarbeitet werden sollen. MUX0 selektiert dabei das erste Analogsignal, MUX1 selektiert das zweite Analogsignal. Für die Herkunft der beiden zu verarbeitenden Analogsignale gibt es drei Möglichkeiten.

Entweder stammt das erste und/oder das zweite Analogsignal vom ABUS, oder das erste und/oder das zweite Analogsignal sind identisch mit der festen Massebezugsspannung GND, oder das erste und/oder das zweite Analogsignal sind identisch mit dem Ausgangssignal der BIPS (0225), welches mittels der Leitung 0252 auf jeweils einen Eingang von MUX0 und MUX1

zurückgeführt wird. Das erste Analogsignal gelangt von MUX0 über die Leitung 0246 auf BUFF0 (0223). Das zweite Analogsignal gelangt von MUX1 über die Leitung 0247 auf BUFF1 (0224). Die beiden Analogsignale können in BUFF0 bzw. BUFF1 gemäß den

5 in Abschnitt über die Ecell beschriebenen Betriebsarten von BUFF0 und BUFF1 verändert werden. BUFF0 und BUFF1 können über die Leitung 0242 unabhängig voneinander durch die CL (0210) gesteuert werden. Das analoge Ausgangssignal von BUFF0 (0223) gelangt über die Leitung 0248 auf den ersten Analogeingang

10 der XAPU (0220). Das analoge Ausgangssignal von BUFF1 (0224) gelangt über die Leitung 0249 auf den zweiten Analogeingang der XAPU (0220). Die XAPU (0220) verarbeitet die beiden analogen Eingangssignale zu einem analogen Ausgangssignal gemäß der durch die CL (0210) über die Leitung 0243 programmierten

15 Funktion, wie in Abschnitt Ecell beschrieben. Das analoge Ausgangssignal der XAPU (0220) wird mittels der Leitung 0250 an eine weitere Speicherstufe (BIPS, 0225) übertragen. Die BIPS der ECELL und die BIPS der SCELL können gleich sein. Die Funktion der BIPS (0225) wird von der CL (0210) durch die

20 Leitung 0244 gesteuert. Das analoge Ausgangssignal der BIPS wird durch die Leitung 0251 auf den Demultiplexer (DeMUX, 0226) übertragen, der das Signal auf den ABUS (0231) aufschaltet. Der DeMUX wird von der CL (0210) gesteuert.

25 Die Logiksektion (0201) der ECELL besteht aus einer vollständigen Logiksektion, wie sie in einer SCELL zu finden ist, also der CL (0210) und der LOGUNIT (0211), die über die Leitung (0240) miteinander in Verbindung stehen. Die Logiksektion der ECELL ist darüberhinaus in der Lage, die

30 XAPU (0120) mit ihrem gegenüber der APU einer SCELL erweiterten Funktionsumfang zu steuern und zu verwalten.

Beispielsweise logische Operationen wie beispielsweise NAND, NOR, AND, OR, XOR ermöglichen. Eingangsvariablen solcher Operationen können solche Kriterien sein, die auch zur Bildung eines RECONREQ-Signals führen, aber auch digitale Signale, die eigens dafür erzeugt werden.

Figur 3 zeigt eine mögliche Art der Realisierung von BUFF0 bzw. BUFF1. OP0 ist ein Operationsverstärker, der so beschaltet ist, daß er das am Eingang IN liegende analoge Signal wahlweise invertiert oder durchschleift. Die Betriebsart wird durch DeMUX0 ausgewählt. Wenn am Steuereingang NONINV INV eine logische 0 liegt, wird das Eingangssignal durchgeschleift, wenn am Steuereingang NONINV INV eine logische 1 liegt, wird das Eingangssignal invertiert. Über DeMUX1 wird entschieden, ob das Signal im Kondensator C zwischengespeichert wird (BUFFER), oder ob es ohne Zwischenspeicherung am Ausgang OUT von OP1 zur Verfügung steht (PASS). Zwischenspeicherung erfolgt, wenn der Steuereingang BUFF PASS eine logische 0 erhält. Keine Zwischenspeicherung erfolgt, wenn der Steuereingang BUFF PASS eine logische 1 erhält.

Figur 4 zeigt, wie beispielsweise der Ausdruck $f(t)^{g(t)}$ berechnet werden kann. Dazu wird in der ersten Zelle $f(t)$ logarithmiert, das bedeutet, von $f(t)$ wird der Logarithmus zur beliebigen, aber festen Basis a gebildet. Dazu kann eine SCELL dienen, die als Logarithmierer konfiguriert ist. Das Ergebnis dieser Operation wird in der zweiten Zelle mit $g(t)$ multipliziert. Dazu kann eine ECELL dienen, die beide Signale nach Art eines spannungsgesteuerten Verstärkers miteinander multipliziert.

In der dritten Zelle wird die Basis a mit dem Ergebnis der Multiplikationsoperation potenziert. Dazu kann eine SCELL dienen, die als Delogarithmierer konfiguriert ist. Das Ergebnis der Delogarithmierungsoperation entspricht dem Ausdruck

5 $([f(t)]^{g(t)})$.

Vorstehend wurde beschrieben, wie eine Einheit mit konfigurierbaren analogen Einheiten aufgebaut werden kann. Es wurde vorgeschlagen, analoge Signale für Arbeiten mit Zellen so

10 auszubilden, dass sie im Betrieb anderer Zellen rekonfigurierbar sind und es wurde vorgeschlagen, ihnen dafür eine geeignete Beschaltung zuzuordnen. Es ist nun einzuschätzen, dass die Möglichkeit besteht, einen Baustein zu bilden, bei welchem eine Signalverarbeitung sowohl analog als auch digital

15 erfolgt. Es ist dann möglich, die digitale Signalverarbeitung gleichfalls mit rekonfigurierbaren Bauelementen vorzusehen, etwa durch ein multidimensionales Feld rekonfigurierbarer digitaler Einheiten, wie es in den verschiedenen Patentanmeldungen des vorliegenden Anmelders

20 beschrieben wurde. Um die erforderliche Wandlung vorzusehen, können einzelne oder mehrere Umsetzerstufen vorgesehen sein, d. h. ein oder mehrere Analog-Digital-Wandler und erforderlichenfalls mehrere Digital-Analog-Wandler. Überdies ist es möglich, verschiedene Wandlerverfahren einzusetzen und die

25 Genauigkeit der Wandlung bei Vorsehen mehrerer Wandlereinheiten unterschiedlich zu gestalten. Gleichfalls ist es möglich, neben einfachen logischen Schaltungen, die einem Analogelement zugeordnet werden, auch komplexere Logik- und Funktionskreise vorgesehen werden können.

30

Es wird einzuschätzen sein, dass die Vielzahl der Analogelemente, Busse usw. sowie der gegebenenfalls erforderlichen

Wandlereinheiten einem jeweiligen Zweck ohne weiteres anpassbar sind, etwa um Hochfrequenzanwendungen zu genügen oder, bei Niederfrequenzanwendungen, eine extreme Rauscharmut bzw. ein sehr gutes Signal- Rauschverhältnis vorzusehen.

5

Es sei weiter erwähnt, daß bevorzugt die digitalen und analogen Elemente gemischt werden, insbesondere auf ein und demselben IC. Dazu kann in einem gemischten Feld mittels eines oder mehrerer ADCs und/oder DACs und/oder Komparatoren
10 ein Übergangsmittel vorgesehen werden. Dies ist vorteilhaft, weil etwa im Bereich Software defined Radio die rein digitale Abarbeitung der ankommenden hochfrequenten schwachen Antennensignale noch problematisch ist, wobei trotzdem auch bezüglich der analogen Signalbe- und/oder -verarbeitung eine
15 große Wahlfreiheit erwünscht ist.

Die Erfindung betrifft weiter Vorrichtungen und Verfahren zur Verbesserung des Transfers von Daten innerhalb von mehrdimensionalen Anordnungen von Sendern und Empfängern bzw. -zellen.
20 Daß diese gerade in kritischen Applikationen wie Software defined Radio besonders relevant ist, sei erwähnt.

Die Zellen von etwa multidimensionalen Prozessorfeldern können nun unterschiedlichen Funktionen ausführen, etwa
25 Bool'sche Verknüpfungen von Eingangs-Operanden bewirken,

Zwischen ihnen verlaufen Verbindungen, die gleichfalls einstellbar sind, typisch etwa Busse, die auf verschiedene Weise eine Vernetzung bewirken können und so ein in seiner Vernetzung einstellbares multidimensionales Feld aufbauen. Über die
30 Busse oder anderen Leitungen tauschen die Zellen miteinander wie erforderlich Informationen aus, etwa Statussignale,

Trigger oder die zu verarbeitenden Daten. Typisch sind dabei in einem zweidimensionalen Prozessorfeld etwa die Zellen Reihen- und spaltenweise angeordnet, wobei die Ausgänge von Zellen einer ersten Reihe auf Busse geführt, an die zugleich die Eingänge der Zellen der nächsten Reihe zu koppeln sind. Bei einer bekannten Anordnung (Pact XPP) sind zudem Vorwärts- und Rückwärtsregister vorgesehen, um Daten unter Umgehung von Zellen auf Bussysteme anderer Reihen zu leiten, ein Balancing von parallel auszuführenden Zweigen zu erreichen, usw. Es ist auch schon vorgeschlagen worden, derartige Vor- und/oder Rückwärtsregister mit einer über den reinen Datentransfer hinausgehenden Funktionalität zu versehen.

Um eine bestimmte Art der Datenverarbeitung durchzuführen, muß jeder Zelle eine bestimmte Funktion zugewiesen werden und es ist eine geeignete Vernetzung vorzusehen. Es muß dazu, bevor das multidimensionale Prozessorfeld Daten wie gewünscht verarbeitet, festgelegt werden, welche Zelle welche Funktion ausführen soll, es ist für jede an einer Datenverarbeitungsaufgabe beteiligten Zelle eine Funktion festzulegen und es muß die Vernetzung bestimmt werden. Dabei ist es wünschenswert, die Funktion und Vernetzung so zu wählen, daß die Datenverarbeitung möglichst zügig erfolgen kann. Oftmals ist es jedoch nicht möglich, eine Konfiguration zu finden, die den gewünschten Datentransfer in optimaler Weise gewährleistet. Es müssen dann suboptimale Konfigurationen verwendet werden.

Wünschenswert ist es hier, eine Möglichkeit zu schaffen, die Konfigurierbarkeit zu erleichtern.

Es wird dabei weiter vorgeschlagen, daß bei einem multidimensionalen Prozessorfeld aufweisend eine Vielzahl benachbart

angeordneter Datenverarbeitungszellen mit Eingängen, die Daten von Vernetzungswegen erhalten, einer Operanden-Verknüpfungseinheit, die diese entsprechend der jeweiligen Funktion ihrer Operanden-Verknüpfungseinheit verknüpfen und
5 Ausgängen, um die Daten verknüpft auf Vernetzungswege aufzugeben, vorgesehen ist, daß die Datenverarbeitungszellen ein Aspektverhältnis aufweisen, das wenigstens 1,5:1, bevorzugt 2:1 beträgt. Dies ermöglicht das bevorzugte Pipelining in den PAEs und/oder den Bussen. Dabei ist es bevorzugt, aber nicht
10 zwingend, in insbesondere jeder PAE ein eigenes Pipelining vorzusehen, was Takterhöhungen ermöglicht

Damit wird eine wesentliche Verbesserung der Verknüpfbarkeit erreicht, ohne daß teure Siliziumfläche für zusätzliche Bus-
15 verbindungen bereitgestellt werden muß oder eine besonders komplexe Topologie gewählt werden muß. Die Verbesserungen der Verbindbarkeit ergeben sich vielmehr allein daraus, daß der Datentransfer quer zu den Zellen verkürzt wird und damit Daten innerhalb kürzerer Zeiten, bezogen auf die zum
20 durchströmen bzw. Verarbeiten in der Zelle erforderlichen Zeiten selbst, von Zelle zu Zelle gelangen. Damit wächst die Anzahl der noch als nächste Nachbarn zu bezeichnenden Zellen, die also noch innerhalb eines Taktes zu erreichen sind. Es ergibt sich etwa bei zweidimensionalen Feldern eine Anord-
25 nung, bei der eine Zelle funktional mehr nächste Nachbarn besitzt, als sich topologisch bei reiner Geometriebetrachtung im zweidimensionalen Fall ergibt. Mit andern Worten ergibt sich nur durch die Veränderung des Aspektverhältnisses funktional eine mehr als zweidimensionale Konnektivität.

30

Bei den Zellen wird es sich insbesondere um PAE-Zellen mit EALU handeln, wie sie per se aus dem vorzitierten Stand der

Technik bekannt sind. Bei solchen Zellen wird es sich wie bevorzugt um grobgranular konfigurierbare Zellen handeln.

Es ist möglich und bevorzugt, wenn die Datenverarbeitungszellen in Reihen und Spalten angeordnet sind. Dies erlaubt eine besonders günstige Auslegung der Zellen, die typisch näherungsweise trapezförmig bzw. rechteckig sind. Es kann dann vorgesehen sein, daß zumindest bei einem Teil der Datenverarbeitungszellen Dateneingänge vorgesehen sind, um von einer oberen Reihe Daten zu erhalten und Datenausgänge, um an eine untere Reihe Daten auszugeben. In einem solchen Fall ergibt sich die verbesserte Konnektivität in beiden Reihen.

Typisch wird es sich um ein Prozessorfeld handeln, bei dem die Datenverarbeitungseinheiten EALUs, ALU und/oder registerflankierte Zellen sind, d.h. es werden zur Verbindung unterschiedlicher Reihen typisch neben den datenverarbeitenden und dabei Daten verzögerungsfrei, d.h. etwa schnellstmöglich weiterleitenden Zellen noch Register vorhanden sein, die gerade dazu dienen, Daten bei der Weiterleitung zu verzögern, sei es, um unkontrollierte Rückkopplungsschleifen zu verhindern bzw. unterbrechen (Prinzip der sog. **Annihilated Feedback Loop Termination**- sog. AFTER-Zellen) oder bei datenaufspaltendem Durchlaufen von Zweigen und nachfolgendem Wiedervereinigen einen zeitlichen Gleichlauf zu erzwingen (Balancing).

Mit einem solchen Prozessorfeld ist es nun möglich, eine Konfiguration derart zu wählen, daß, wenn Zellen für die Konfiguration ausgewählt und in Funktion und Vernetzung bestimmt werden, wobei eine Vernetzung derart bestimmt wird, daß Daten von Zelle zu Zelle zumindest weitgehend verzöge-

rungsfrei übertragbar sind, vorgesehen ist, daß als benachbarte Zellen, zwischen denen Daten binnen eines Taktes oder einer geringen Taktzahl übertragbar sind, auch solche berücksichtigt werden, die nicht unmittelbar nebeneinander liegen, sondern in der Breite durch eine Strecke getrennt sind, die geringer ist als die Länge der Zelle. Daß eine Heruntertaktung der Zellen im Vergleich zu den Bussen per se möglich ist, sei als bevorzugt offenbart. Offensichtlich kann aber auch in Ausnahmefällen eine in die andere Richtung gehende Taktabweichung erfolgen oder auf eine solche verzichtet werden.

Es sei darauf hingewiesen, daß das angegebene minimale Aspektverhältnis das wenigstens 1,5:1 beträgt, bevorzugt noch größere Werte annimmt und sich bei sorgfältiger Auslegung der Einheiten durchaus im Bereich zwischen 5:1 und 10:1 bewegen kann.

Die Erfindung wird im folgenden anhand der Zeichnung beschrieben, worin gezeigt ist durch

Fig. C1 ein Prozessorfeld der vorliegenden Erfindung

Nach Fig. 1 umfaßt ein allgemein mit 1 bezeichnetes Prozessorfeld 1 eine Vielzahl benachbart angeordneter Datenverarbeitungszellen 2 mit Eingängen 3, die Daten von Vernetzungswegen 4 erhalten, einer Operanden-Verknüpfungseinheit 5, die diese entsprechend der jeweiligen Funktion ihrer Operanden-Verknüpfungseinheit 5 verknüpfen und Ausgängen 6, um die Daten verknüpft auf Vernetzungswege 4 aufzugeben, wobei die Datenverarbeitungszellen bzw. ihre da-

tendurchflossene Operanden-Verknüpfungseinheit 5 ein Aspektverhältnis von Länge zu Breite aufweisen, das größer ist als 2:1 beträgt.

5 Bei dem Prozessorfeld 1 handelt es sich vorliegend um eine per se als XPP bekannte Anordnung; alternativ kann es als ein Array aus zur Laufzeit partiell rekonfigurierbaren Elementen angeordnet sein können, etwa als Prozessor, Koprozessor, DSP, usw.. Das Prozessorfeld ist im dargestellten Fall aus 3 Reihen
10 hen und 4 Spalten aufgebaut, aber nur aus Gründen der Übersichtlichkeit so vergleichsweise klein gewählt. Typisch wird es größer ausgelegt werden.

Die Datenverarbeitungszellen 2 sind grobgranular konfigurierbar
15 bar und weisen feingranulare Statemachines auf. Sie sind auf per se bekannte Weise rekonfigurierbar, ohne den Betrieb zu stören. Auf die hier realisierte, aber nicht näher zu erläuternde Möglichkeit der zentralen Konfigurationsvorgabe etwa durch einen Konfigurationsmanager, der Waverekonfiguration
20 etc. sei hingewiesen. Die Zellen enthalten als Operanden-Verknüpfungseinheit 5 eine ALU-Einheit, in der arithmetische Operationen wie Addition, Multiplikation, Subtraktion und Division an bis zu drei eingehenden Operanden durchgeführt werden können, sowie Verknüpfungen wie ISTgrößer? ISTkleiner?
25 ISTNull? sowie XOR, OR, AND NAND etc. Die ALU-Einheit ist mittig angeordnet und flankiert von einem Vorwärts- und einem Rückwärtsregister, die in per se bekannter Weise über die Anschlüsse der Datenverarbeitungszelle 2 gleichfalls mit den Vernetzungswegen 4 verbunden werden können.

30

Die Datenein- und ausgänge 3 bzw 6 sind über Multiplexer mit den Verbindungswegen 4 verbunden. Es ist im vorliegenden Fall

ein Bussystem mit einer Vielzahl von Leitungen vorgesehen, um die Zellen in den Reihen und Spalten konfigurierbar miteinander zu vernetzen.

- 5 Das Aspektverhältnis der ALU-Einheit beträgt nun im dargestellten Beispiel 6:1, d.h. die Zelle ist sehr viel länger als breit.

Die Anordnung wird nun verwendet wie folgt:

10

- Es wird zunächst ein Programm zur Ausführung auf dem Array 1 ausgewählt. Dann wird mit per se bekannten Mitteln eine Konfiguration bestimmt, die einen optimalen Daten-Durchsatz erlaubt. Hierbei wird nun berücksichtigt, daß Daten auch an
15 Zellen, die nicht unmittelbar in der Reihe darunter oder seitlich neben einer gegebenen Zelle liegen, sondern z.B. drei Spalten seitlich versetzt sind, innerhalb eines Verarbeitungstaktes Daten erhalten können, ohne daß größere Verzögerungen auftreten. Die unter Berücksichtigung dieser
20 erweiterten Nächste-Nachbar-Definition erhaltene Konfiguration wird auf das Array aufkonfiguriert und ausgeführt.

- Die vorliegende Erfindung befasst sich aber nicht nur mit dem
25 vorteilhaften Aufbau eines multidimensionalen Feldes rekonfigurierbarer Elemente wie bei rekonfigurierbaren Prozessoren, sondern auch mit Verfahren zu deren Betrieb, etwa derart, daß eine Übersetzung einer klassischen Hochsprache (PROGRAMM) wie Pascal, C, C++, Java etc. auf eine rekonfigurierbare Archi-
30 tektur ermöglicht wird.

Oftmals wird hier nicht das gesamte multidimensionale Feld rekonfigurierbarer Elemente samt aller zwischen den Daten handhabenden Elementen vorgesehenen Bussystemen, Verbindungsleitungen usw. zur Umkonfiguration freigegeben, sondern es besteht vielmehr das Erfordernis, eine neue Aufgabe einem kleinen Teilbereich des multidimensionalen Feldes zuzuordnen. Es ist überdies häufig nicht vorhersagbar, wie dieser Teilbereich beschaffen sein wird. Dies gilt insbesondere dann, wenn auf dem multidimensionalen Feld rekonfigurierbarer Elemente mehrere Aufgaben parallel abgearbeitet werden müssen, etwa im Wege des Multitasking, und/oder nicht vorhergesagt werden kann, wann dort, etwa bei Echtzeitanwendungen, welche Ressourcen zwecks Umkonfiguration freigegeben werden.

Prinzipiell besteht die Möglichkeit, einen Code, der auf dem multidimensionalen Feld rekonfigurierbarer Elemente abgearbeitet werden soll, zur Laufzeit zu übersetzen, also erst dann, wenn die Abarbeitung anderer Aufgaben schon begonnen hat, festzulegen, wie der als nächstes auszuführende Code bestimmten rekonfigurierbaren Elementen zuzuordnen ist, wie die Verbindung zwischen diesen laufen soll, welche Zwischenspeicherungen erforderlich sind usw. Es ist einsichtig, dass eine solche Vorgehensweise zur Übersetzung einen vergleichsweise hohen momentanen Datenverarbeitungsaufwand erfordert. Gerade in kritischen Rechnerapplikationen, die ein Höchstmaß an Rechenleistung erfordern, ist es gewünscht, für eine solche Übersetzung während der Laufzeit keine zusätzliche Rechenleistung zu verbrauchen. Es ist daher auch schon üblich, Programmcode vor Beginn des Programmes zu kompilieren und dann Teilkonfigurationen zu bestimmen, die jeweils in das Feld hinein konfiguriert werden, sobald dort entsprechende Ressourcen frei sind.

Ein Problem besteht allerdings darin, dass, gerade bei Echtzeitanwendungen, im Vorfeld nicht feststeht, wie die jeweils verfügbaren Ressourcen angeordnet sind. Dies betrifft einerseits die Funktionalität der zur Datenhandhabung verfügbaren Elemente, in die hinein konfiguriert werden könnte, sofern nicht alle Daten handhabenden Elemente dieselbe Funktion besitzen. So wäre denkbar, in einem multidimensionalen Feld rekonfigurierbarer Elemente verschiedene Zellen mit Rechenwerken auszustatten, die für Fließkomma-Berechnungen ausgelegt sind, Elemente vorzusehen, die lediglich Bool'sche Daten handhaben, Elemente, die über zugeordnete Speicher verfügen, Elemente, mit oder in denen Sequenzer vorgesehen werden können usw. Hier ist eine Ausführung mit Vorkompilierung darauf angewiesen, entweder mit der Umkonfiguration zu warten, bis genau jene Zellen zur Verfügung stehen, die die in der Vorkompilierung festgelegten Funktionen und Anordnungen besitzen. Auch muß bei der Vorkompilierung zudem der kleinste, allen Zellen gemeinsame Funktionsumfang verwendet werden. Beides verschwendet Ressourcen. Zudem ist meist nicht klar, wie die für die Umkonfiguration freigegebenen Elemente angeordnet sind und welche Verbindungen verfügbar sind; auch dadurch wird gegebenenfalls die Hineinkonfiguration einer neuen Aufgabe (Task) massiv erschwert.

Das Problem wird noch gravierender, wenn große Bereiche des multidimensionalen Feldes freigegeben werden und prinzipiell die Möglichkeit und/oder der Zwang besteht, mehrere Konfigurationen für unterschiedliche Aufgaben gleichzeitig in das Feld hinein zu konfigurieren.

Es wird somit gemäß einem ersten wesentlichen Aspekt der Erfindung ein Verfahren zum Betrieb eines multidimensionalen Feldes rekonfigurierbarer Elemente vorgeschlagen, worin Gruppen zusammen datenhandhabender Elemente in vorbestimmter

5 Weise während der Laufzeit zur Abarbeitung vorgegebener Aufgaben im Feld konfiguriert werden und wobei vorgesehen ist, dass für zumindest eine abzuarbeitende Aufgabe eine Mehrzahl von solchen Elementgruppenanordnungen im multidimensionalen Feld bestimmt wird, die zur Abarbeitung der vorgegebenen Auf-
 10 gabe geeignet sind, für die Abarbeitung der vorgegebenen Aufgabe eine dann besonders geeignete Elementgruppenanordnung aus der Mehrzahl ausgewählt wird und die ausgewählte Anordnung in das Feld hinein konfiguriert wird.

15 Die Erfindung schlägt somit vor, bei der Vorbereitung der eigentlichen Datenverarbeitung eine Vielzahl von Anordnungen bzw. Konfigurationen vorherzubestimmen und dann aus den vorbestimmten Elementgruppenanordnungen eine solche auszuwählen, die für die Abarbeitung der vorgegebenen Aufgabe bei den dann
 20 gegebenen Feldressourcen besonders gut geeignet ist. Damit ist eine wesentliche Verbesserung im Betrieb eines multidimensionalen Feldes rekonfigurierbarer Elemente im Wesentlichen durch eine einfache Erweiterung des Compilers gegeben, mit dem die zuvor programmierten Codes übersetzt
 25 werden, und zwar dadurch, dass dieser nicht nur eine einzige Konfiguration für eine gegebene Aufgabe bestimmt, sondern mehrere solcher Konfigurationen und somit ausnützt, dass keine eindeutige Lösung für das Problem besteht, ein Stück
 30 gegebenen Hochsprachencode auf ein multidimensionales Feld rekonfigurierbarer Elemente zu übersetzen. Es sei erwähnt, daß hier der Begriff "Compiler" verwendet wird für ein Mittel, das Konfickurationen bestimmt, unabhängig davon, ob es sich

um einen Routerteil, einen Übersetzerteil oder einen anderen Teil eines Mittels zur Konfigurationsbestimmung anhand von Programmcodes handelt. Dieses Mittel kann mittels Festverdrahtung, d. h. als Hardware oder als Softwareprogramm
5 realisiert sein.

Es ist möglich, aus dieser Vielzahl von potentiell möglichen Konfigurationen, die zur Abarbeitung eines gegebenen Stück-Codes möglich sind, eine Auswahl anhand der Geometrie zu
10 treffen, die diese Elementgruppenanordnung im Vergleich zu jener besitzt, die die im multidimensionalen Feld für die Umkonfiguration verfügbaren bzw. vermutlich bald verfügbar werdenden Elemente besitzen. So kann durch einen einfachen Mustervergleich versucht werden, eine Konfiguration, d. h.
15 Elementgruppenanordnung, auszuwählen, die möglichst alle der frei gewordenen oder frei werdenden Elemente abdeckt bzw. möglichst wenig Elemente des multidimensionalen Feldes ungenutzt lässt. Wenn lediglich auf die Geometrie Rücksicht genommen wird, etwa weil alle Daten handhabenden Elemente des
20 multidimensionalen Feldes den für die Hineinkonfiguration erforderlichen Funktionsumfang besitzen, so kann die Auswahl mit per se bekannten Algorithmen wie bei der Schnittmusteroptimierung erfolgen. Es kann dabei entweder auf die bereits verfügbaren Elemente Bezug genommen werden oder es kann, insbesondere im Hinblick darauf, dass die Umkonfiguration
25 oftmals die Übertragung von Konfigurationsdaten zu den Elementen umfasst und eine solche Umkonfigurationsdatenübertragung Zeit in Anspruch nimmt, vorgesehen werden, dass auch vermutlich bald verfügbare Elemente
30 mit bei der Auswahl der jeweils optimalen Geometrie berücksichtigt werden. Dabei kann ausgenutzt werden, dass es häufig möglich ist, vorherzusagen, dass bestimmte Elemente bald für

die Umkonfiguration verfügbar werden, etwa wenn sie Daten für die Weiterverarbeitung von Zellen erhalten haben, die bereits ihre Umkonfigurierbarkeit angedeutet haben und die Anzahl der noch erforderlichen Verarbeitungstakte, der hierzu daten-

5 stromabwärts liegenden Zellen endlich und abschätzbar oder bekannt ist. Derartige Information ist erfindungsgemäß als Umkonfigurierbarkeits-Vorhersage verwaltbar. Daß zu den verfügbaren und/oder benötigten Elementen auch Busverbindungen, Leitungen etc. zählen, sei erwähnt.

10

Die Auswahl der optimalen Konfiguration kann dabei in einem Präprozessor oder einem Teilbereich des multidimensionalen Feldes der rekonfigurierbaren Elemente erfolgen und insbesondere von einem Datenverarbeitungsprogramm und/oder -mittel

15 übernommen werden, das die Durchführung der verschiedenen Aufgaben zeitlich koordiniert, Priorisierungen vornimmt usw. Es kann sich hierbei insbesondere um einen Teil eines Betriebssystems handeln, sofern das multidimensionale Feld rekonfigurierbarer Elemente als Prozessor oder Koprozessor

20 ausgeführt ist. Die Verwendbarkeit der CT, eines Schedulers für Hyperthreading, Multitasking, Multithreading usw. sei erwähnt. Auf entsprechende weitere Teile der vorliegenden Anmeldung sei diesbezüglich verwiesen. Daß solche Einheiten hard- und/oder softwaremäßig implementierbar sind, sei er-

25 wähnt.

Gerade dann, wenn Konfigurationsdaten aus einem Speicher eingelesen werden, der nicht vernachlässigbare Zugriffszeiten besitzt, bzw. wenn sie, sollte eine Echtzeitbestimmung einer

30 Konfiguration gewünscht werden, mit nicht vernachlässigbaren Generierungszeiten generiert werden, ist es wünschenswert, zunächst einen Kenndatensatz vorzusehen, der gegenüber dem

eigentlichen Konfigurationsdatensatz in der Grösse reduziert ist und dann eine Auswahl nur auf Grund dieses Kenndatensatzes zu treffen. So kann etwa beim Laden einer neuen Konfiguration aus einem langsamen Speicher wie von einer

5 Festplatte zunächst lediglich ein Kenndatensatz bzw. eine Kenndatensatzgruppe heruntergeladen werden, der die Umrisse der Konfiguration betrifft. Da ein solcher Umrisskenndatensatz in der Grösse gegenüber dem vollständigen Konfigurationsdatensatz typisch sehr reduziert sein wird, ist

10 es auch möglich, eine Vielzahl von Kenndatensätzen für eine Vielzahl unterschiedlicher Konfigurationen vorab in einen Hauptspeicher zu laden, der einen sehr schnellen Zugriff erlaubt, an Hand der unterschiedlichen Konfigurationsdatensätze eine schnelle Auswahl zu treffen und dann aus dem langsamen

15 Speicher die kompletten Konfigurationsdaten zur ausgewählten Konfiguration herunter zu laden. Es sei darauf hingewiesen, dass es in solchen Fällen auch möglich ist, einen Teil der Konfigurationen vorneweg miteinzulesen, etwa dann, wenn absehbar ist, dass bestimmte Konfigurationen typisch bevorzugt

20 sind, sei es, weil statistische Auswertungen des typischen Datenverarbeitungsbetriebes für eine Vielzahl von multidimensionalen Feldern rekonfigurierbarer Elemente oder für ein einzelnes multidimensionales Feld dies ergeben haben, etwa weil zu einer Gruppe von Anwendungen wie im UMTS-Base-

25 Station-Bereich durch Analyse typischer Aufgaben festgestellt wurde, dass bestimmte Umkonfigurationen besonders häufig auftreten, oder weil für einen einzelnen Benutzer festgestellt wurde, dass immer wieder dieselben Anwendungen parallel in bestimmter Weise in das Feld hinein konfiguriert werden müssen.

30 Das Vorab-Mitladen bestimmter Konfigurationen kann auch dann sinnvoll sein, wenn sich diese Konfigurationen durch eine besonders einfache Geometrie auszeichnet, etwa weil sehr

kleine Volumina des multidimensionalen Feldes rekonfigurierbarer Elemente davon überdeckt werden (Volumina nimmt hier auf das Volumen des multidimensionalen Feldes Bezug, bezeichnet also bei zweidimensionalen Felder rekonfigurierbarer
 5 Elemente die Fläche bzw. Flächengeometrie der für die Umkonfiguration verfügbaren rekonfigurierbaren Elemente etc.).

Es ist auch möglich und insbesondere bei der Abarbeitung komplexer Aufgabenstellungen, sei es durch die Abarbeitung
 10 besonders rechenintensiver Probleme, sei es im Multitasking, Multithreading oder bei anderen Formen der Parallelabarbeitung von Daten, auch bevorzugt, zu überprüfen, ob mehrere, insbesondere gleich priorisierte Elementgruppenanordnungen für unterschiedliche Aufgaben simultan durch geeignete Auswahl
 15 in das Feld hinein konfigurierbar sind. Dabei kann, abhängig von der Priorisierung einer bestimmten Aufgabe vorgesehen werden, dass die für die Abarbeitung einer vorgegebenen Aufgabe zur Verfügung gestellte Fläche oder Abarbeitungszeit größer oder kleiner ausfällt, etwa indem durch
 20 Aufbau von Sequenzern mit den Daten handhabenden Elementen die Größe einer Konfiguration unter Verlangsamung der Datenabarbeitung verringert wird.

Es kann auch erwünscht sein, dass zunächst eine erste Elementgruppenanordnung in das Feld hinein konfiguriert wird und
 25 begonnen wird, mit dieser Elementgruppenanordnung die Aufgabe abzuarbeiten, bis ein vorgegebenes Ereignis auftritt und dann unter zumindest partieller Rekonfiguration mit der Aufgabenabarbeitung in einer weiteren Elementgruppenanordnung
 30 fortgefahren wird. Hierbei kann etwa vorgesehen werden, dass zur Erzielung einer bevorzugten Geometrie von Konfigurationen auf dem multidimensionalen Feld, etwa streifenförmig hinter-

einander angeordneten Zellen für jede Aufgabe, zu festgelegten Taktzeiten, etwa alle tausend, zehntausend oder hunderttausend Takte, die Abarbeitung aller oder eines Teiles aller Konfigurationen unterbrochen wird, die Ergebnisse wie
 5 erforderlich zwischengespeichert werden, auch was nur in einer Konfiguration intern erforderliche Daten wie Schleifen-, Zählerstände usw. angeht, und dann eine Neukonfiguration mit entsprechenden, bevorzugten Geometrien vorgenommen wird, um so ein allmähliches Zerfasern von Konfigurationen zu vermei-
 10 den, was schon aufgrund des erhöhten Bedarfs an Busleitungen unerwünscht ist.

Alternativ und/oder zusätzlich ist es auch möglich, selbstfaltende Konfigurationen vorzusehen, wobei etwa zunächst mit
 15 der Abarbeitung einer Konfiguration über das gesamte Array begonnen wird, und sich dann, sobald zusätzliche Ressourcen durch eine andere Aufgabe angefordert werden, diese erste Konfiguration mehr oder minder selbsttätig zusammenzieht, etwa durch Sequenzerbildung mit einem Element, um Elemente für
 20 die neue Aufgabe freizugeben. Dieses Zusammenziehen kann durch Vorgabe neuer, Platz sparenderer Konfigurationen für ein und dieselbe Aufgabe erreicht werden, insbesondere dann, wenn diese Platz sparenderen Konfigurationen in bei den Daten handhabenden Elementen vorgesehenen Konfigurationsspeichern
 25 mit abgelegt werden. Auf die Anmeldung zur Wave-Rekonfiguration sei dabei nur beispielsweise hingewiesen. Es ergibt sich dann eine Situation, bei der Konfigurationen allmählich enger und enger zusammenrücken.

30 Die Auswahl einer vorgegebenen Elementgruppenanordnung, die in ein Feld hinein zu konfigurieren ist, kann, abgesehen von der verfügbaren Geometrie, auch von anderen Parametern abhän-

gig gemacht werden. Hierzu zählt unter anderem die erzielbare Abarbeitungsgeschwindigkeit, die Priorität einer Aufgabe und/oder der Energieverbrauch, der zur Abarbeitung einer vorgegebenen Aufgabe in einer vorgegebenen Zeit erforderlich

5 ist. Es sei darauf hingewiesen, dass mehrere Parameter gleichzeitig betrachtet werden können, sei es dadurch, dass zunächst anhand eines ersten Parameters wie des benötigten Feldvolumens als gleichwertig angesehene Konfigurationen durch Betrachten eines zweiten Parameters verworfen werden, sei es, indem, etwa mit Methoden der unscharfen Logik, mehrere Parameter gleichzeitig soweit wie möglich optimiert werden.

Die Erfindung wird nun im Folgenden nur beispielsweise anhand 15 der Figuren erläutert, worin gezeigt ist durch

Fig. D1 ein multidimensionales Feld Daten handhabender Elemente in teilweise zu rekonfigurierendem Zustand;

20 Fig. D2 Beispiele für unterschiedliche Konfigurationsgeometrien;

Fig. D3 ein teilweise zur Laufzeit umkonfigurierter Prozessor.

25 Nach Fig. 1 umfasst eine allgemein mit 1 bezeichnete Datenverarbeitungsvorrichtung 1 ein multidimensionales Feld rekonfigurierbarer Elemente 2 sowie einen Präprozessor 3, der Konfigurationen in das multidimensionale Feld 1 über geeignete Datenbusse 4 einspeist und Informationen über 30 rekonfigurierbare Elemente aus dem multidimensionalen Feld 2 mehrere Elemente erhält sowie mit einem Speicher 5 mit langsamem Zugriff verbunden ist, in dem Konfigurationen für in

dem multidimensionalen Feld 2 abzuarbeitende Aufgaben vorab abgelegt sind.

Der multidimensionale Prozessor 1 ist im vorliegenden Beispiel eine XPU-Architektur, die PAE als konfigurierbare Elemente aufweist und nach PACT02, 04, 08, 10, 13 aufgebaut ist. Er erhält Daten von Eingabe-/Ausgabe-Schnittstellen 6 in Echtzeit zur Abarbeitung, wobei nicht vorhersehbar ist, wie diese Daten eintreffen und/oder abzuarbeiten sind. Es können hierfür eine Tastatur, Bilder abrufende Kameras, A/D-Wandler usw. vorgesehen sein.

Das multidimensionale Feld 2 besteht, zur Vereinfachung der Veranschaulichung, obwohl technologisch keinesfalls zwingend, vorliegend nur aus einer Reihe ausschließlich identischer datenhandhabender Elemente, zwischen denen geeignete Vernetzungen über Busse und dergleichen konfigurierbar sind. Aus Gründen der Anschaulichkeit wird vorliegend von unbegrenzten Busressourcen ausgegangen, obgleich rein praktisch die typische Anwendung auch solcher Ressourcen und deren Knappheit bei der Vorabbestimmung multipler Konfigurationsmöglichkeiten berücksichtigt wird. Die datenhandhabenden Elemente sind im vorliegenden Fall geeignet, sequenzartig die Befehle abzuarbeiten, wie dies per se bekannt ist, das heißt, es ist möglich, Sequenzen über einzelne Zellen oder Gruppen derer aufzubauen. Daß hierbei ein Zeiteilungsmultiplexing möglich ist, sei erwähnt. Dies erlaubt eine entsprechende Faltung mehrerer Operationen, dann dann bei großen Arrays bzw. mehr Platz auch ausgefaltet werden können.

Das multidimensionale Feld 2 ist laufzeitrekonfigurierbar, das heißt es ist möglich, einzelnen der datenhandhabenden

Elemente oder Gruppen derer neue Aufgaben während der Laufzeit zuzuordnen, ohne den Betrieb der Gesamt-Anordnung oder anderer Elemente bzw. Gruppen derer insgesamt zu unterbrechen. Wie bevorzugt und per se bekannt sind den

5 datenhandhabenden Elementen Konfigurationsspeicher lokal zugeordnet, genau so wie Register, nämlich Forward- und Backward-Register, Busleitungen, feingranulare Zustandsmaschinen zum Austausch von Triggersignalen untereinander und mit der Präprozessoreinheit 3 usw. Auf die Möglichkeit der
10 Ausgestaltung der rekonfigurierbaren Elemente nach PCT-DE 97/02949, PCT-DE 97/02998, PCT-DE 98/00334, PCT/DE 99/00504, PCT/DE 99/00505, PCT/DE 00/01869 usw. sei hingewiesen. Die vorgenannten Schutzrechte und die weiteren Schutzrechte des Anmelders zu rekonfigurierbaren Prozessoren, deren Teilen und
15 Verfahren zu deren Betrieb sind zu Offenbarungszwecken vollumfänglich integriert.

Der Präprozessor 3 ist dazu ausgebildet, Konfigurationen in das multidimensionale Feld hinein zu laden und zwar über Leitungen 4, wenn er aus dem multidimensionalen Feld die Meldung erhält, dass einzelne Elemente oder Gruppen derer rekonfigurierbar sind. Der Präprozessor 3 enthält einen lokalen Speicher (Cache) und ist mit einem weiteren Speicher 5 (Harddisk, RAM) verbunden, auf den langsamer zugegriffen werden
25 kann und auf dem Konfigurationsdaten abgelegt sind. Geeignet ist etwa eine CT.

Es sei darauf hingewiesen, dass es nicht erforderlich ist, den Präprozessor 3 als externes Bauteil vorzusehen. Die gezeigte Darstellung wurde ausschließlich aus didaktischen
30 Gründen gewählt. Er kann integriert sein mit dem multidimensionalen Feld 2 auf einem einzelnen Chip und/oder seine

Funktion kann durch einzelne Daten handhabende Elemente 2 des Prozessorfeldes ausgeführt werden.

Über die Leitungen 4 werden Konfigurationsdaten und Konfigurationsanforderungen übertragen. Auf die Implementierung von Rdy/Ack-Protokollen, Vorab-Konfiguration von Elementen in elementennahen Speichern etc., die möglich, aber nicht zwingend ist, wird hingewiesen.

10 Im Speicher 5 sind nun eine Vielzahl von Konfigurationen für unterschiedliche Aufgaben und Kenndaten hierzu abgelegt. Dies wird für ein einfaches Beispiel mit Bezug auf Fig. 2 erläutert.

15 Nach Fig. 2 sind etliche Konfigurationen für zwei Aufgaben a) und b) abgelegt. Wie ersichtlich, sind für die Aufgabe a) insgesamt vier Konfigurationen abgelegt, die alle dieselbe Funktion ausführen, aber unterschiedliche Verbindungen der Zellen untereinander haben und sich insbesondere hinsichtlich
20 ihrer äußeren geometrischen Gestalt unterscheiden, in der die Zellen angeordnet sind.

Wie ersichtlich, sind beispielsweise drei Konfigurationen vorabgelegt, in denen sieben datenhandhabende Elemente wie
25 PAEs benötigt werden und eine Konfiguration, in der unter Ausnutzung der Sequenzereigenschaft der datenhandhabenden Elemente nur vier Elemente benötigt werden. Die geometrische Form der jeweiligen Konfiguration ist dabei wie durch die Zahlen in Klammern angedeutet, gleichfalls mitabgelegt. Dieser
30 Kenndatensatz umfasst eine erste Ziffer, die angibt, wieviele Spalten Abstand die äußersten Zellen links und rechts voneinander haben; es folgen dann nach einem Komma die

Anzahl der Elemente in einer Spalte, die benötigt werden. Sind in einer Spalte Reihen frei, d. h. nicht belegt, steht gleichfalls ein b in der Kennung. Ist dabei eine Spalte freigelassen, das heißt von der jeweiligen Konfiguration bis auf
 5 Busse nicht belegt, so steht hierfür ein b in der Konfiguration. Dies ist ersichtlich an den Konfigurationen I und II. Die Daten zu einer Spalte sind von jenen der nächsten Spalte durch Komma getrennt. Ähnliche Konfigurationsdaten sind auch für eine zweite Konfiguration abgelegt b).

10

Die Anordnung wird verwendet wie folgt:

Wenn im multidimensionalen Feld rekonfigurierbarer Elemente Ressourcen für die Rekonfiguration freigegeben sind, wie dies
 15 durch die "0" in Fig. 2 dargestellt ist, so lädt der Präprozessor 3 aus dem Speicher 5 zunächst die wenig umfangreichen und damit schnell zu ladenden Kennsätze zu den Konfigurationen. Er bestimmt dann, welche Aufgabe schnell abzuarbeiten ist und welche Konfigurationen gemeinsam besonders gut in das
 20 Feld eingeladen werden können. Dies geschieht durch Vergleich der maximalen Spaltenbreite einer möglichen Konfiguration mit der tatsächlich verfügbaren Spaltenbreite. Zur Aufgabe a) können so Konfiguration III und IV verworfen werden, die zuviel Spalten benötigen. Aus den verbleibenden sind aufgrund
 25 der geometrischen Form die Konfigurationen I und II gleichfalls zu verwerfen. Es wird dann untersucht, welche Konfiguration von b) zu laden wäre. Hier sind per se alle drei Konfigurationen ladbar.

30 Nun wird untersucht, ob es eine Möglichkeit gibt, von den verbleibenden Konfigurationen zu den Aufgaben gleichzeitig zwei Konfigurationen in das Feld zu laden. Dazu werden die

Konfigurationen in unterschiedlicher Weise aneinander gesetzt und es wird die benötigte maximale Spalten- und Reihenzahl mit der verfügbaren Maximalzahl verglichen. Es wird so festgestellt, daß sich eine optimale Ausnutzung der

5 freigewordenen Elemente ergibt, wenn die Konfiguration Ib und die Konfiguration Ia unmittelbar übereinander angeordnet werden. Diese Konfigurationen werden dann in das Prozessorfeld geladen.

10 Daraufhin kann die Datenverarbeitung mit einer Konfigurationsanordnung wie in Fig. 3 gezeigt fortgesetzt werden. Es sei erwähnt, dass in Fällen, in denen unterschiedliche Daten handhabende Elemente vorgesehen werden, die entsprechende Information gleichfalls im Kenndatensatz abgelegt werden
15 kann.

Wie vorstehend ersichtlich, ist die Art und Weise, wie ein gegebenes Prozessorfeld für eine vorgegebene Weise konfiguriert werden muß, nicht eindeutig. Dies gilt insbesondere
20 dann, wenn es sich um komplexe Felder handelt, bei denen etwa zumindest zum Teil in den Leitungen Register vorgesehen sind und weiter mit diesen insbesondere auch Additionen und/oder Vergleiche von Daten durchzuführen sind, wie dies auch in arithmetische Logikeinheiten (ALUs) aufweisenden Logikzellen
25 des Feldes der Fall sein kann. Oft ist es auch möglich und/oder nötig, etwa bei Inbetriebnahme, aus vielen Konfigurationen mehrere mögliche zu wählen.

Es ist bereits vorgeschlagen worden, eine Konfiguration aus
30 mehreren per se verwendbaren auszuwählen anhand der momentanen Konfigurierbarkeit unter geometrischen Aspekten, der Ressourcenverfügbarkeit und/oder auszuwählen aufgrund von Ge-

schwindigkeitsaspekten. Dies kann die Auswahl erleichtern, stellt aber oftmals nur unzureichende Kriterien dar. Es ist wünschenswert, die Konfigurationsauswahl weiter verbessern zu können. Es ist weiter oftmals möglich, eine bestimmte Daten-

5 verarbeitungsaufgabe selbst auf unterschiedliche Weise durchzuführen. So ist etwa eine Reihe von Algorithmen bekannt, die es erlauben, einen Satz Daten auf unterschiedliche Weise zu sortieren. Auch hier ist es erforderlich, zwischen unterschiedlichen Algorithmen, die prinzipiell geeignet sind,

10 eine bestimmte Datenverarbeitungsaufgabe zu bewältigen, aufgrund von objektivierbaren Kriterien zu wählen. Daß diese Wahl zur Laufzeit und/oder davor erfolgen kann, sei erwähnt. Insgesamt ist es somit wünschenswert, Auswahlmöglichkeiten bei der Datenverarbeitung mit konfigurierbaren multidimensio-

15 nalen Prozessorfeldern zu verbessern, etwa um bei festgespeicherten Konfigurationen sicherzustellen, daß eine bereits anwendungszweckoptimierte Auswahl getroffen wurde.

Die vorliegende Erfindung schlägt somit in einem ersten

20 Grundgedanken ein Verfahren zur Auswahl eines aus einer Vielzahl von Wegen zur Erzielung eines Datenverarbeitungsergebnisses bei der Datenverarbeitung unter zumindest möglicher Verwendung multidimensionaler Felder konfigurierbarer Datenhandhabungselemente vor, bei welchem

25 vorgesehen ist, daß den Datenhandhabungselementen konfigurationsabhängig verbrauchsbezogene kennzeichnende Größen zugeordnet werden und eine Wegauswahl anhand der Zuordnung erfolgt.

30 Ein weiterer Grundgedanke kann somit in der Erkenntnis gesehen werden, daß bestimmten Datenverarbeitungswegen typische Leistungs- bzw. Energieaufnahmewerte zugeordnet werden kön-

nen, um dann eine Wegauswahl unter Berücksichtigung derselben vorzunehmen. Als Erzielung eines Datenverarbeitungsergebnisses wird auch eine bestimmte Art und Weise einer Zwischenergebnisberechnung und/oder einer Datenhandhabung etc. betrachtet. Durch die Zuordnung verbrauchskennzeichnender Größen wird also eine wesentliche Objektivierung der Wegauswahl möglich.

Die Auswahl eines Weges kann beispielsweise die Wahl eines gegebenen Algorithmus aus einer Vielzahl unterschiedlicher Algorithmen umfassen, sei es für Aufgaben wie das Sortieren von Daten, bestimmte mathematische Transformationen oder dergleichen. Liegen in einer Programmmodul-Bibliothek etwa mehrere Sortieralgorithmen, Algorithmen zur Bestimmung einer Fourier-Transformierten oder dergleichen vor, so kann z. B. zu jeder eine verbrauchskennzeichnende Größe ermittelt werden und es wird dann unter Berücksichtigung derselben eine Auswahl vorgenommen. So wird es möglich, Algorithmen zu wählen, die beispielsweise einen besonders niedrigen Energieverbrauch besitzen. Dies kann für mobile Anwendungen wie bei Laptops, Funktelefonen und dergleichen sinnvoll sein, bietet aber auch in Bereichen Vorteile, bei denen sehr rechenintensive Aufgaben zu bewältigen sind, etwa in Servern, Basestations usw., wo die in einer Verarbeitungseinheit erzeugte Leistung weggekühlt und/oder abgeführt werden muß. Durch die Erfindung können so Systemgesamtkosten minimiert werden. Auch kann etwa ein Place&Route-Algorithmus die Optimierung ausnutzen, etwa zur Erzielung niederenergetischer Anordnungen. Dies ist besonders bevorzugt und wird als für sich erfinderisch angesehen.

Es ist weiter möglich, für ein und denselben Algorithmus eine Vielzahl unterschiedlicher Konfigurationen vorzusehen, etwa unter Berücksichtigung verschiedener, simultan und/oder sequenziell auf das multidimensionale Feld zu konfigurierender
 5 Teilaufgaben, und dann eine Auswahl aus diesen unter Auswertung der jeweils zugeordneten Größe vorzunehmen.

Weiter ist es möglich, unter Anwendung des erfindungsgemäßen Verfahrens darüber zu befinden, ob eine gegebene Aufgabe der
 10 Datenverarbeitung, bzw. eine Teilaufgabe, dem betrachteten multidimensionalen Feld konfigurierbarer Datenhandhabungselemente und/oder einem anderen Element zur Datenverarbeitung außerhalb des multidimensionalen Feldes zugeordnet werden soll; so kann etwa darüber entschieden werden, ob dann, wenn
 15 das multidimensionale Feld als Koprozessor dient, eine bestimmte Teilaufgabe besser auf einer rein sequenziell arbeitenden CPU oder dem rekonfigurierbaren, typisch als Datenflußprozessor oder dergleichen arbeitenden multidimensionalen Feld abgearbeitet werden soll. Es ist auch
 20 möglich, über das Erfordernis oder die Sinnhaftigkeit dedizierter Schaltkreise wie ASICs für bestimmte Aufgaben zu befinden.

Typisch wird es sich bei dem Feld konfigurierbarer Datenhandhabungselemente um ein zweidimensionales Feld handeln. Es sei
 25 erwähnt, daß die Erfindung für Felder wie FPGAs, XPP-Prozessoren usw. anwendbar ist. Es ist besonders bevorzugt für zur Laufzeit konfigurierbare, insbesondere partiell zur Laufzeit ohne Störung nicht umzukonfigurierende Elemente teil-rekonfigurierbarer Prozessorfelder.
 30

In typischen Anwendungen wie bei XPP-Feldern werden als zu berücksichtigende Datenhandhabungselemente insbesondere zu-
mindest einige, bevorzugt alle der Elemente Busse, Register,
ALUs, RAMs, I/O-Ports sowie konfigurierende Einheiten (CT's)
5 erfaßt. Es sei darauf hingewiesen, daß von bestimmten dieser
Teile nur eine abgeschätzte oder partielle Verbrauchsberück-
sichtigung erforderlich ist; so brauchen etwa bei Bussen
lediglich bestimmte Treiberstufen und dergleichen berücksich-
tigt werden. Weiter kann es erforderlich sein, auch
10 Taktbeschaltungen mit zu erfassen, sei es, weil in bestimmten
Datenverarbeitungswegen eine Voll- oder Teilabschaltung eines
Taktzweiges möglich ist, sei es, weil bestimmte Schaltungsbe-
reiche mit einem unterschiedlichen Takt versorgt werden
können oder müssen.

15

Es ist bevorzugt, wenn der kennzeichnende Wert nur grob abge-
schätzt wird, etwa dahingehend, daß festgestellt wird, ob ein
bestimmtes Element momentan verwendet wird und/oder konfigu-
riert ist, oder ob es stattdessen nicht verwendet wird und
20 gegebenenfalls sogar von einer Spannungsversorgung bis auf
einen Wake-up-Schaltkreis und/oder von einer Taktversorgung
zumindest weitgehend abgeklemmt ist. Es ist also nicht erfor-
derlich, eine absolut exakte Verbrauchskennzeichnung
vorzunehmen, etwa unter Festlegung des Verbrauchs der spezi-
25 fischen algebraischen Operation, die einer jeweiligen
arithmetisch-logischen Einheit aktuell und/oder dauerhaft zu-
geordnet wird. Vielmehr kann es ausreichend sein, die
verbrauchskennzeichnende Größe nur dahingehend zu bestimmen,
ob und wie weit tatsächlich das jeweilige Element im Augen-
30 blick verwendet wird. Ausnahmen hiervon sind möglich. Eine
Ausnahme kann insbesondere gemacht werden für Operationen wie
Multiplikationen, bei denen sehr große Schaltkreisflächen mit

Leistung versorgt werden müssen. In einem solchen Fall kann eine weitere Detaillierung vorgesehen werden.

Es ist möglich und bevorzugt, jedem unterschiedlichen Daten-
5 handhabungselement unterschiedliche Kennzahlen, wie strom-
und/oder leistungsaufnahmebezogener Größen als verbrauchs-
kennzeichnende Größen zuzuordnen. Dieses kann gegebenenfalls
in taktabhängiger Weise (Leistungsaufnahme pro Taktfrequenz)
geschehen. Weiter ist es möglich, eine Auswahl unter Berück-
10 sichtigung eines Summenwertes vorzunehmen, also über die
Betrachtung des Gesamtverbrauchs oder abgeschätzten Gesamt-
verbrauchs eines betrachteten Weges zu entscheiden.

Die Auswahl wird typisch nicht alleine unter Berücksichtigung
15 der verbrauchskennzeichnenden Größen erfolgen, sondern kann
andere Parameter miterfassen, etwa eine erforderliche Ausführ-
ungszeit, erforderliche Ressourcen auf einem
multidimensionalen Feld, eine aktuelle oder erwartete Prozes-
sorauslastung durch andere Aufgaben und/oder eine aktuell
20 gewünschte beziehungsweise erwartete oder zulässige Lei-
stungsaufnahme. Die Kennzahlen sind durch Meßwerte und/oder
Hardware- bzw. Syntheseanalysen erhältlich und insbesondere
in Look-Up-Tabellen hinterlegbar.

25 Die Auswahl des jeweiligen Weges kann vor der eigentlichen
Datenverarbeitung erfolgen, etwa bei der Festlegung von spä-
ter zu ladenden Konfigurationen unter mehreren, theoretisch
implementierbaren Konfigurationen. In einem solchen Fall ist
es besonders ist bevorzugt, wenn die kennzeichnende Größe
30 während der Simulation der Datenverarbeitungsfunktionen mit
ermittelt wird. Alternativ kann vorgesehen werden, daß die
Auswahl unter verschiedenen möglichen Wegen während der Lauf-

zeit erfolgt. In einem solchen Fall werden etwa mehrere mögliche Algorithmen, z. B. um Daten zu sortieren, bereitgestellt, es wird dann abgefragt, wie viele einzelne Daten zu sortieren sind und wie gegebenenfalls der Ordnungsgrad dieser Daten ist und es wird erst danach eine Auswahl unter verschiedenen, vorher bestimmten Algorithmen anhand der ihnen zugeordneten parametrisierten Verbrauchskennzeichnungsgrößen wie der Gesamtleistungsaufnahme usw. vorgenommen. Analog kann auch eine Konfiguration zur Laufzeit abhängig von z. B. einer momentan möglichen oder gewünschten Stromaufnahme erfolgen.

Dieser Aspekt der Erfindung wird im folgenden nur beispielsweise beschrieben, ohne daß Bezug genommen wird auf eine

Figur.

Es wird zunächst eine gewünschte Art der Daten-Bearbeitung festgelegt, die auf dem Prozessorfeld durchgeführt werden soll. Beispielsweise wird ein Viterbi-Algorithmus programmiert und eine für das betrachtete Prozessorfeld geeignete Konfiguration bestimmt. Es wird dann ermittelt, welche Einheiten auf dem Prozessorfeld verwendet werden und über wie viele Takte dies geschieht. Bei der Betrachtung der verwendeten Elemente werden in einem Beispiel berücksichtigt ALUs, Vorwärts- und Rückwärts-Register (FREG und BREG) sowie Schalter in Bussen (LSW und RSW). Es wird dann der gesamte Energieverbrauch je Elementart bestimmt und anschließend der gesamte Energieverbrauch aller unterschiedlichen Einheiten. Die Energieverbrauchswerte für ein einzelnes Element je Takt sind ihrerseits abgeschätzt aus Simulationen der Hardware-

schaltungen bei der betrachteten Architektur und sind für das Verfahren der Erfindung tabellarisch hinterlegt.

In dem betrachteten praktischen Beispiel sind für die Implementierung eines gegebenen Viterbi-Algorithmus 10 ALUs, 17 Vorwärtsregister, 23 Rückwärtsregister sowie 30 Busschalter (LSW) in einer und 35 Schalter in der entgegengesetzten Richtung (RSW) erforderlich. Bei einem Energieverbrauch von 4,85 pW/Hz je ALU, 7,01 pW/Hz pro FREG, 7,02 pW/Hz je BREG sowie 2,03 pW/Hz je Busschalter ergibt sich folgende Tabelle:

Number of cycles: 1582

Energy consumption

15

			Einzelkennzahl		Gesamtkennzahl
	ALU:	10,00	x	4,85	= 48,50
	FREG:	17,00	x	7,01	= 119,17
20	BREG:	23,00	x	7,02	= 161,46
	LSW:	30,00	x	2,03	= 60,90
	RSW:	35,00	x	2,03	= 71,05

	Total:				461,08 pW/Hz

25

Es kann nun der Implementierung der Viterbi-Transformation ein Gesamtleistungsverbrauch von 461,08 pW/Hz zugeordnet werden und dieser so erhaltene Wert verglichen werden mit Werten, die bei anderen Algorithmen und/oder Konfigurationen erhalten werden und/oder durch dedizierte Schaltkreise wie ASICs zu erhalten sind.

30

Es sei nun erwähnt, daß die Auswahl einer aus einer Vielzahl von Konfigurationen, auch dann sinnvoll sein kann, wenn das Datenverarbeitungslogikzellen- und/oder, hier gleichbedeutend, Mischfeld aus analogen und/oder digitalen Zellen (wie
5 beschrieben) an eine insbesondere sequentielle CPU angekoppelt ist.

Ein Problem bei herkömmlichen Ansätzen zu rekonfigurierbaren Technologien besteht dann aber oft, wenn die Datenverarbeitung primär auf einer sequenziellen CPU unter Hinzuziehung
10 eines konfigurierbaren Datenverarbeitungslogikzellenfeldes oder dergleichen erfolgen soll und/oder eine Datenverarbeitung gewünscht ist, in der viele und/oder umfangreiche sequenziell auszuführende Verarbeitungsschritte vorliegen.

15

Es sind Ansätze bekannt, die sich damit befassen, wie eine Datenverarbeitung sowohl auf einem konfigurierbaren Datenverarbeitungslogikzellenfeld als auch auf einer CPU erfolgen kann.

20

So ist aus der WO 00/49496 ein Verfahren zum Ausführen eines Computerprogrammes mit einem Prozessor bekannt, der eine konfigurierbare funktionelle Einheit umfaßt, die in der Lage ist, rekonfigurierbare Anweisungen auszuführen, deren Effekt
25 zur Laufzeit durch Laden eines Konfigurationsprogrammes redefiniert werden kann, wobei das Verfahren die Schritte umfaßt, daß Kombinationen rekonfigurierbarer Anweisungen ausgewählt, ein respektives Konfigurationsprogramm für jede Kombination erzeugt und das Computerprogramm ausgeführt wird.
30 Dabei soll jedes Mal, wenn eine Anweisung aus einer der Kombinationen während der Ausführung gebraucht wird und die konfigurierbare funktionelle Einheit nicht mit dem Konfigura-

tionsprogramm für diese Kombination konfiguriert ist, das Konfigurationsprogramm für alle der Anweisungen der Kombination in die konfigurierbare funktionelle Einheit geladen werden. Weiter ist aus der WO 02/50665 A1 eine Datenverarbeitungsvorrichtung mit einer konfigurierbaren funktionellen Einheit bekannt, wobei die konfigurierbare funktionelle Einheit dazu dient, eine Anweisung gemäß einer konfigurierbaren Funktion auszuführen. Die konfigurierbare funktionelle Einheit weist eine Vielzahl von unabhängigen konfigurierbaren Logikblöcken zum Ausführen programmierbarer Logikoperationen auf, um die konfigurierbare Funktion zu implementieren. Konfigurierbare Verbindungsschaltkreise sind zwischen den konfigurierbaren Logikblöcken und sowohl den Eingängen als auch den Ausgängen der konfigurierbaren funktionellen Einheit vorgesehen. Dies erlaubt eine Optimalisierung der Verteilung von Logikfunktionen über die konfigurierbaren Logikblöcke.

Ein Problem bei herkömmlichen Architekturen besteht auch dann, wenn eine Ankopplung erfolgen soll und/oder Technologien wie Datastreaming, Hyperthreading, Multithreading und so weiter in sinnvoller und Performance steigernder Weise ausgenutzt werden sollen. Die beispielhaft erwähnte Technologie der vorzitierten Nicht-Anmelder-Dokumente zeigt etwa eine Anordnung, bei der zwar Konfigurationen in ein konfigurierbares Datenverarbeitungslogikzellenfeld geladen werden können, bei welchen allerdings der Datenaustausch zwischen der ALU der CPU und dem konfigurierbaren Datenverarbeitungslogikzellenfeld, sei es ein FPGA, DSP oder dergleichen, über die Register erfolgt. Mit anderen Worten müssen Daten aus einem Datenstrom zunächst sequenziell in Register geschrieben werden und dann sequenziell wieder in diesen abgelegt werden. Auch ist ein Problem dann gegeben, wenn ein Zugriff auf Daten

von extern erfolgen soll, da selbst dann noch Probleme beim zeitlichen Ablauf der Datenverarbeitung im Vergleich zur ALU und bei der Zuweisung von Konfigurationen und so weiter bestehen. Die herkömmlichen Anordnungen, wie sie aus den Nicht-
 5 Anmelder-eigenen Schutzrechten bekannt sind, werden unter anderem dazu verwendet, Funktionen im konfigurierbaren Datenverarbeitungslogikzellenfeld, DFP, FPGA oder dergleichen abzuarbeiten, die nicht effizient auf der CPU-eigenen ALU abzuarbeiten sind. Damit wird das konfigurierbare
 10 Datenverarbeitungslogikzellenfeld praktisch verwendet, um benutzerdefinierte Opcodes zu ermöglichen, die eine effizientere Abarbeitung von Algorithmen ermöglichen, als dies auf dem ALU-Rechenwerk der CPU ohne konfigurierbare Datenverarbeitungslogikzellenfeldunterstützung möglich wäre.

15

Im Stand der Technik ist, wie erkannt wurde, die Ankopplung demnach im Regelfall wortbasiert, nicht jedoch blockbasiert, wie es zur datenströmenden Verarbeitung erforderlich wäre. Es ist zunächst wünschenswert, eine effizientere Datenverarbeitung zu ermöglichen, als dies mit einer engen Ankopplung über
 20 Register der Fall ist.

Eine weitere Möglichkeit zur Verwendung von Logikzellenfeldern aus grob- und/oder feingranular gebauten Logikzellen und
 25 Logikzellenelementen besteht in einer sehr losen Ankopplung eines solchen Feldes an eine herkömmliche CPU und/oder ein CPU-Kern bei eingebetteten Systemen. Hierbei kann ein herkömmliches, sequenzielles Programm auf einer CPU oder dergleichen laufen, beispielsweise ein in C, C++ oder dergleichen geschriebenes Programm, wobei von diesem Aufrufe
 30 einer Datenstromverarbeitung auf dem fein- und/oder grobgranularen Datenverarbeitungslogikzellenfeld instantiiert

werden. Problematisch ist dann, dass beim Programmieren für dieses Logikzellenfeld ein nicht in C oder einer anderen sequenziellen Hochsprache geschriebenes Programm für die Datenstromabarbeitung vorgesehen werden muss. Erwünscht wäre

5 hier, dass sowohl auf der herkömmlichen CPU-Architektur als auch auf einem mit diesen gemeinsam betriebenen Datenverarbeitungslogikzellenfeld C-Programme oder dergleichen abzuarbeiten sind, das heißt, dass insbesondere mit dem Datenverarbeitungslogikzellenfeld in quasi sequenzieller

10 Programmabarbeitung dennoch eine Datenstromfähigkeit erhalten bleibt, während simultan auch insbesondere möglich bleibt, dass ein CPU-Betrieb in nicht zu loser Ankopplung möglich ist. Es ist auch bereits bekannt, innerhalb einer Datenverarbeitungslogikzellenfeldanordnung, wie sie insbesondere aus

15 PACT02 (DE 196 51 075.9-53, WO 98/26356), PACT04 (DE 196 54 846.2-53, WO 98/29952), PACT08, (DE 197 04 728.9, WO 98/35299) PACT13 (DE 199 26 538.0, WO 00/77652) PACT31 (DE 102 12 621.6-53, PCT/EP 02/10572) bekannt ist, auch eine sequenzielle Datenverarbeitung innerhalb des

20 Datenverarbeitungslogikzellenfeldes vorzusehen. Hierbei wird dann allerdings innerhalb einer einzelnen Konfiguration, beispielsweise um Ressourcen zu sparen, eine Zeitoptimierung zu erzielen und so weiter, eine partielle Abarbeitung erzielt, ohne dass diese bereits dazu führt, dass ein Programmierer

25 ein Stück Hochsprachencode automatisch leicht ohne weiteres auf ein Datenverarbeitungslogikzellenfeld umsetzen kann, wie dies bei herkömmlichen Maschinenmodellen für sequenzielle Prozessoren der Fall ist. Die Umsetzung von Hochsprachencode auf Datenverarbeitungslogikzellenfelder nach Prinzipien der

30 Modelle für sequenziell arbeitende Maschinen ist weiterhin schwierig.

Aus dem Stand der Technik ist weiter bekannt, dass mehrere Konfigurationen, die eine jeweils unterschiedliche Funktionsweise von Arrayteilen bewirken, simultan auf dem Prozessorfeld (PA) abgearbeitet werden können und dass ein

5 Wechsel von einer oder einigen der Konfiguration(en) ohne Störung anderer zur Laufzeit erfolgen kann. Es sind Verfahren und in Hardware implementierte Mittel zu deren Umsetzung bekannt, wie sichergestellt werden kann, dass dabei ein

10 Abarbeiten von auf das Feld zu ladenden Teilkonfigurationen ohne Deadlock erfolgen kann. Verwiesen wird hierzu insbesondere auf die die Filmo-Technik betreffenden Anmeldungen PACT05 (DE 196 54 593.5-53, WO 98/31102) PACT10 (DE 198 07 872.2, WO 99/44147, WO 99/44120) PACT13 (DE 199 26 538.0, WO 00/77652), PACT17 (DE 100 28 397.7, WO 02/13000). Diese Technologie ermöglicht in gewisser Weise bereits eine

15 Parallelisierung und, bei entsprechender Gestaltung und Zuordnung der Konfigurationen, auch eine Art Multitasking/Multithreading und zwar dergestalt, dass eine Planung, das heißt ein Scheduling und/oder eine Zeitnutzungsplanungssteuerung vorgesehen ist. Es sind also aus dem Stand der Technik schon Zeitnutzungsplanungssteuerungsmittel und -verfahren per se bekannt, die, zumindest unter entsprechender

20 Zuordnung von Konfigurationen zu einzelnen Aufgaben und/oder Fäden zu Konfigurationen und/oder Konfigurationsfolgen, ein Multitasking und/oder Multithreading erlauben. Die Verwendung solcher Zeitnutzungsplanungssteuermittel, die im Stand der Technik zur Konfigurierung und/oder Konfigurationsverwaltung verwendet wurden, zu Zwecken des Scheduling von Tasks, Threads, Multi- und Hyperthreads wird per se als erfinderisch

25 angesehen.

30

Wünschenswert ist auch zumindest gemäß einem Teilaspekt in bevorzugten Varianten, moderne Technologien der Datenverarbeitung und Programmabarbeitung wie Multitasking, Multithreading, Hyperthreading unterstützen zu können, zumindest in bevorzugten Varianten einer Halbleiterarchitektur.

Ein weiterer wesentlicher Aspekt der vorliegenden Erfindung ist somit darin zu sehen, dass dem Datenverarbeitungslogikzellenfeld Daten im Ansprechen auf die Ausführung einer Ladekonfiguration durch das Datenverarbeitungslogikzellenfeld zugeführt werden und/oder Daten aus diesem Datenverarbeitungslogikzellenfeld weggeschrieben (STORE) werden, indem eine STORE-Konfiguration entsprechend abgearbeitet wird. Diese Lade- und oder Speicherkonfigurationen sind dabei bevorzugt derart auszugestalten, dass innerhalb des Datenverarbeitungslogikzellenfeldes und/oder einer anderen Einheit wie einer RISC-Architektur direkt oder indirekt Adressen jener Speicherstellen generiert werden, auf welche ladend und/oder speichernd direkt oder indirekt zugegriffen werden soll. Es ist durch diese Einkonfiguration von Adressgeneratoren innerhalb einer Konfiguration möglich, eine Vielzahl von Daten in das Datenverarbeitungslogikzellenfeld einzuladen, wo sie gegebenenfalls in internen Speichern (iRAM) ablegbar sind und/oder wo sie in internen Zellen wie EALUs mit Registern und/oder dergleichen eigenen Speichermitteln abgelegt werden können. Die Lade- beziehungsweise Speicherkonfiguration ermöglicht somit ein blockweises und nahezu datenstromartiges, insbesondere gegenüber Einzelzugriff vergleichsweises schnelles Laden von Daten und es kann eine solche Lade-Konfiguration ausgeführt werden vor einer oder mehreren tatsächlich Daten auswertend und/oder verändernd abarbeitenden Konfiguration(en), mit welcher/n die vorab geladenen Daten verarbeitet

werden. Das Datenladen kann dabei typisch bei großen Logikzellenfeldern in kleinen Teilbereichen derselben geschehen, während andere Teilbereiche mit anderen Aufgaben befaßt sind. Bei der in anderen veröffentlichten Dokumenten des Anmelders

5 beschriebenen Ping-Pong-artigen Datenverarbeitung, bei der auf beiden Seiten eines Datenverarbeitungsfeldes Speicherzellen vorgesehen sind, wobei die Daten in einem ersten Verarbeitungsschritt von dem Speicher auf der einen Seite durch das Datenverarbeitungsfeld zum Speicher auf der anderen

10 Seite strömen, dort die beim ersten Felddurchströmen erhaltenen Zwischenergebnisse im zweiten Speicher abgelegt werden, gegebenenfalls das Feld umkonfiguriert wird, die Zwischenergebnisse dann für die Weiterverarbeitung zurückströmen usw., kann etwa eine Speicherseite durch eine LOAD-Konfiguration in

15 einem Array-Teil mit neuen Daten vorgeladen werden, während aus der gegenüberliegenden Speicherseite Daten mit einer STORE-Konfiguration in einem anderen Array-Teil weggeschrieben werden. Dieses simultane LOAD/STORE-Vorgehen ist im übrigen auch ohne räumliche Speicherbereichstrennung möglich.

20 Das Laden kann insbesondere aus einem Cache und in diesen hinein erfolgen. Dies hat die Vorteile, dass die externe Kommunikation mit größeren Speicherbänken über den Cachecontroller gehandhabt wird, ohne dass innerhalb des Datenverarbeitungslogikzellenfeldes separate Schaltanordnungen dafür vorgesehen sein müssen, dass der Zugriff in lesender oder schreibender Weise bei Cache-Speichermitteln typisch

25 sehr schnell und mit allenfalls geringer Latenzzeit erfolgen wird und dass auch typisch eine CPU-Einheit, dort typisch über eine separate LOAD/STORE-Einheit, an diesen Cache angebunden ist, sodass ein Zugriff auf Daten und ein Austausch derselben zwischen CPU-Kern und Datenverarbeitungslogikzel-

30

lenfeld blockweise schnell und derart erfolgen kann, dass nicht für jedes Übergeben von Daten ein separater Befehl etwa aus dem OpCode-Fetcher der CPU abgeholt und verarbeitet werden muss.

5

Es erweist sich diese Cacheankoppelung auch als wesentlich günstiger als eine Ankopplung eines Datenverarbeitungslogikzellenfeldes an die ALU über Register, wenn diese Register nur über eine LOAD/STORE-Einheit mit einem Cache kommunizieren, wie dies aus den Nicht-PACT-eigenen zitierten Schriften per se bekannt ist.

10

Es kann eine weitere Datenverbindung zu der Lade/Speicher-einheit der oder einer dem Datenverarbeitungslogikzellenfeld zugeordneten Sequenziell-CPU-Einheit vorgesehen sein und/oder zu deren Register.

15

Es sei erwähnt, dass ein Ansprechen derartiger Einheiten über separate Eingangs-/Ausgangsanschlüsse (IO-Ports) der insbesondere als VPU beziehungsweise XPP ausgestaltbaren Datenverarbeitungslogikzellenanordnung erfolgen kann und/oder durch einen oder mehrere einem Einzelport nachgeschaltete Multiplexer.

20

Dass neben dem insbesondere blockweisen und/oder streamenden und/oder im Random-Access, insbesondere im RMW-Modus (Read-Modify-Write-Modus) erfolgenden Zugriff auf Cache-Bereiche in schreibender und/oder lesender Weise und/oder die LOAD/STORE-Einheit und/oder die (per se im Stand der Technik bekannte) Verbindung mit dem Register der Sequenziell-CPU auch eine Verbindung mit einem externen Massenspeicher wie einem RAM, einer Festplatte und/oder einem anderen Datenaustauschport

30

wie einer Antenne und so weiter erfolgen kann, sei auch erwähnt. Es kann für diesen Zugriff auf Cache- und/oder LOAD/STORE-Einheit- und/oder registereinheitverschiedene Speichermittel ein separater Port vorgesehen sein. Dass hier

- 5 geeignete Treiber, Signalaufbereiter für Pegelanpassung und so weiter vorgesehen sein können, sei erwähnt. Im Übrigen sei erwähnt, dass insbesondere, jedoch nicht ausschließlich zur Aufbereitung eines in das Datenverarbeitungslogikzellenfeld hineinströmenden oder in diesem strömenden Datenstrom die Logikzellen des Feldes ALUs bzw. EALUs umfassen können und
- 10 typisch werden, denen eingangs- und/oder ausgangsseitig, insbesondere sowohl eingangs- als auch ausgangsseitig kurze, feingranular konfigurierbare, FPGA-artige Schaltkreise vorgesetzt sein können, um etwa aus einem kontinuierlichen
- 15 Datenstrom Vierbitblöcke herauszuschneiden, wie dies für die MPEG-4-Dekodierung erforderlich ist. Es ist dies zum einen vorteilhaft, wenn ein Datenstrom in die Zelle hineingelangen soll und dort ohne Blockierung von größeren PAE-Einheiten einer Art Vorverarbeitung zu unterwerfen ist. Dies ist auch
- 20 dann von ganz besonderem Vorteil, wenn die ALU als SIMD-Rechenwerk ausgestaltet wird, wobei dann ein sehr breites Dateneingangswort von zum Beispiel 32 Bit Datenbreite über die vorgeschalteten zB FPGA-artigen Streifen aufgespalten wird in mehrere parallele Datenwörter von zum Beispiel 4 Bit Breite,
- 25 die dann in den SIMD-Rechenwerken parallel abgearbeitet werden können, was die Gesamtperformance des Systems signifikant zu erhöhen vermag, sofern entsprechende Anwendung benötigt werden. Es sei darauf hingewiesen, dass vorstehend von FPGA-artigen vor- beziehungsweise nachgeschalteten Strukturen die
- 30 Rede war. Mit FPGA-artig muss aber, was explizit erwähnt sei, nicht zwingend Bezug genommen sein auf 1-Bit-granulare Anordnungen. Es ist insbesondere möglich, statt dieser

hyperfeingranularen Strukturen lediglich feiner granulare Strukturen von zum Beispiel 4 Bit Breite vorzusehen. Das heißt, die FPGA-artigen Eingangs- und/oder Ausgangsstrukturen vor und/oder nach einer insbesondere als SIMD-Rechenwerk ausgestalteten ALU-Einheit sind so konfigurierbar, dass immer 4 Bit breite Datenwörter zugeführt und/oder verarbeitet werden. Es ist möglich, hier eine Kaskadierung vorzusehen, so dass zum Beispiel die einkommenden 32 Bit breiten Datenwörter in 4 separierte bzw. separierende 8-Bit-FPGA-artige, nebeneinander angeordnete Strukturen strömen, diesen 4 Stück 8 Bit breiten FPGA-artigen Strukturen ein zweiter Streifen mit 8 Stück 4 Bit breiten FPGA-artigen Strukturen nachgesetzt ist, und gegebenenfalls nach einem weiteren derartigen Streifen dann, sofern dies für den jeweiligen Zweck als erforderlich erachtet wird, zum Beispiel 16 Stück parallel nebeneinander angeordnete 2 Bit breite FPGA-artige Strukturen vorgesehen werden. Wenn dies der Fall ist, kann gegenüber rein hyperfeingranular FPGA-artigen Strukturen eine beträchtliche Verringerung des Konfigurationsaufwandes erzielt werden. Dass dies überdies dazu führt, dass der Konfigurationsspeicher und so weiter der FPGA-artigen Struktur wesentlich kleiner ausfallen kann und somit eine Einsparung an Chipfläche erzielt wird, sei erwähnt.

Prinzipiell sind die vorstehend beschriebenen Kopplungsvorteile bei Datenblockströmen über den Cache prinzipiell erreichbar; besonders bevorzugt ist es jedoch, wenn der Cache streifenweise (slice-artig) aufgebaut ist und dann ein Zugriff auf mehrere der Slices simultan erfolgen kann, insbesondere auf alle Slices gleichzeitig. Dies ist dann vorteilhaft, wenn, was noch erörtert werden wird, auf dem Datenverarbeitungslogikzellenfeld (XPP) und/oder der Sequen-

ziell-CPU und/oder den Sequenziell-CPU's eine Vielzahl von Threads abzuarbeiten sind, sei es im Wege des Hyperthreadings, des Multitaskings und/oder des Multithreadings. Es sind also bevorzugt Cachespeichermittel mit Scheibenzugriff bzw. Scheibenzugriffsermöglichungssteuermitteln vorgesehen. Es kann dabei z. B. jedem Thread eine eigene Scheibe zugeordnet werden. Dies ermöglicht es später, beim Abarbeiten der Threads sicherzustellen, dass jeweils auf die entsprechenden Cachebereiche bei Wiederaufnahme der mit dem Thread abarbeitenden Befehlsgruppe zugegriffen wird.

Es sei noch einmal erwähnt, dass der Cache nicht zwingend in Slices unterteilt sein muss, und dass, wenn dies der Fall ist, nicht zwingend jeder Slice einem eigenen Thread zugewiesen werden muss. Es sei allerdings darauf hingewiesen, dass dies die bei weitem bevorzugte Methode ist. Es sei weiter darauf hingewiesen, dass es Fälle geben kann, in denen nicht alle Cache-Bereiche simultan oder zu einer gegebenen Zeit temporär benützt werden. Vielmehr ist zu erwarten, dass bei typischen Datenverarbeitungsanwendungen, wie sie in handgehaltenen mobilen Telefonen (Handys), Laptops, Kameras und so weiter auftreten werden, häufig Zeiten vorliegen werden, in denen nicht der gesamte Cache benötigt wird. Es ist daher besonders bevorzugt, wenn einzelne Cache-Bereiche von der Leistungsverorgung derart trennbar sind, dass ihr Energieverbrauch signifikant absinkt, insbesondere auf oder nahe null. Dies kann bei sliceweiser Ausgestaltung des Caches durch sliceweise Abschaltung derselben über geeignete Leistungsabtrennmittel geschehen. Die Abtrennung kann entweder über eine Heruntertaktung, Taktabtrennung oder eine Leistungsabtrennung erfolgen. Es kann insbesondere einer einzelnen Cache-Scheibe oder dergleichen eine Zugriffserken-

nung zugeordnet sein, welche dazu ausgebildet ist, zu erkennen, ob ein jeweiliger Cache-Bereich beziehungsweise eine jeweilige Cache-Scheibe momentan einen ihm zugeordneten Thread, Hyperthread oder Task hat, von welchem er benützt wird. Sofern dann vom Zugriffserkennungsmittel festgestellt wird, dass dies nicht der Fall ist, wird typisch eine Abtrennung vom Takt und/oder sogar der Leistung möglich sein. Es sei erwähnt, dass bei Wiedereinschalten der Leistung nach einem Abtrennen ein sofortiges Wiederansprechen des Cachebereiches möglich ist, also keine signifikante Verzögerung durch das An- und Ausschalten der Leistungszufuhr zu erwarten ist, sofern mit gängigen geeigneten Halbleitertechnologien eine Implementierung in Hardware erfolgt.

Ein weiterer besonderer Vorteil, der sich bei der vorliegenden Erfindung ergibt, besteht darin, dass zwar eine besonders effiziente Kopplung bezüglich des Übertrags von Daten beziehungsweise Operanden in insbesondere blockweiser Form gegeben ist, dass aber dennoch ein Balancing nicht in der Weise erforderlich ist, dass die exakt gleiche Verarbeitungszeit in Sequenziell-CPU und XPP beziehungsweise Datenverarbeitungslogikzellenfeld erforderlich ist. Vielmehr erfolgt die Verarbeitung in einer praktisch oftmals unabhängigen Weise, insbesondere derart, dass die Sequenziell-CPU und die Datenverarbeitungslogikzellenfeldanordnung für einen Scheduler oder dergleichen als separate Ressourcen betrachtbar sind. Dies erlaubt eine sofortige Umsetzung bekannter Datenverarbeitungsprogrammaufspaltungstechnologien wie Multitasking, Multithreading und Hyperthreading. Der sich ergebende Vorteil, dass ein Pfadbalaancing nicht erforderlich ist, führt dazu, dass beispielsweise in der Sequenziell-CPU beliebige Anzahlen von Pipelinestufen durchlaufen werden können, Tak-

tungen in unterschiedlicher Weise mögliche sind und so weiter. Ein weiterer Vorteil der vorliegenden Erfindung besteht darin, dass durch das Hineinkonfigurieren einer Ladekonfiguration beziehungsweise einer Storekonfiguration in das XPP

5 oder andere Datenverarbeitungslogikzellenfelder die Daten in das Feld mit einer Geschwindigkeit hineingeladen werden oder aus diesem herausgeschrieben werden können, die nicht mehr bestimmt ist durch die Taktgeschwindigkeit der CPU, die Geschwindigkeit, mit welcher der OpCode-Fetcher arbeitet, oder
 10 dergleichen. Mit anderen Worten ist die Ablaufsteuerung der Sequenziell-CPU nicht mehr flaschenhalsartig begrenzend für den Datendurchsatz des Datenzellenlogikfeldes, ohne dass eine nur noch lose Ankopplung besteht.

15 Während es in einer besonders bevorzugten Variante der Erfindung möglich ist, die für eine XPP-Einheit bekannte CT (bzw. CM; Konfigurationsmanager bzw. Konfigurationstabelle) zu verwenden, um sowohl das Konfigurieren eines oder mehrerer, auch hierarchisch mit mehreren CTs angeordneter XPP-Felder und
 20 gleichzeitig eines oder mehrerer Sequenziell-CPU's, dort quasi als Hyperthreading-Hardwareverwaltung /Scheduler zu verwenden, was den inhärenten Vorteil hat, daß bekannte Technologien wie FILMO usw. für die hardwareunterstützte Verwaltung beim Hyperthreading einsetzbar werden, ist es
 25 alternativ und/oder, insbesondere in hierarchischer Anordnung, zusätzlich möglich, dass ein Datenverarbeitungslogikzellenfeld wie eine XPP Konfigurationen vom OpCode-Fetcher einer Sequenziell-CPU über das Koprozessor-Interface erhält. Dies führt dazu, daß von der
 30 Sequenziell-CPU und/oder einer anderen XPP ein Aufruf instantiiert werden kann, der zu einer Datenabarbeitung auf der XPP führt. Die XPP wird dabei dann z. B. über die beschriebene

Cache-Ankopplung und/oder mittels LOAD- und/oder STORE-Konfigurationen, die Adressgeneratoren für Laden und/oder Wegschreiben von Daten im XPP- bzw. Datenverarbeitungslogikzellenfeld vorsehen, im Datenaustausch gehalten. Mit

5 anderen Worten wird eine Koprozessor-artige Ankopplung eines Datenverarbeitungslogikzellenfeldes möglich, während gleichzeitig ein datenstromartiges Datenladen durch Cache- und/oder I/O-Port-Kopplung erfolgt.

10 Es sei erwähnt, daß die Koprozessor-Ankopplung, d. h. die Ankopplung des Datenverarbeitungslogikzellenfeldes typisch dazu führen wird, daß das Scheduling auch für dieses Logikzellenfeld auf der Sequenziell-CPU oder einer dieser übergeordneten
Schedulereinheit bzw. einem entsprechenden Scheduler
15 erfolgen wird. In einem solchen Fall findet praktisch die Threading-Kontrolle und -verwaltung auf dem Scheduler bzw. der Sequenziell-CPU statt. Obwohl dies per se möglich ist, wird dies, zumindest bei einfachster Implementierung der Erfindung, nicht zwingend der Fall sein. Vielmehr kann eine
20 Verwendung des Datenverarbeitungslogikzellenfeldes durch Aufruf in herkömmlicher Weise wie bei einem Standard-Koprozessor etwa bei 8086/8087-Kombinationen erfolgen.

Weiter sei erwähnt, daß es in einer besonders bevorzugten Variante, unabhängig von der Art der Konfiguration, sei es über
25 das Koprozessor-Interface, den als Scheduler mitdienenden Konfigurationsmanager (CT) der XPP bzw. des Datenverarbeitungslogikzellenfeldes oder dergleichen oder auf andere Weise, möglich ist, im bzw. unmittelbar am Datenverarbeitungslogikzellenfeld bzw. unter Verwaltung des
30 Datenverarbeitungslogikzellenfeldes Speicher, insbesondere interne Speicher, insbesondere bei der XPP-Architektur, wie

sie aus den diversen Voranmeldungen und den Veröffentlichungen des Anmelders bekannt ist, RAM-PAEs, oder andere entsprechend verwaltete oder interne Speicher wie ein Vektorregister anzusprechen, d. h. die über die LOAD-Konfiguration
5 eingeladenen Datenmengen vektorartig wie in Vektorregistern in die internen Speicher abzulegen, dann, nach Umkonfigurieren der XPP bzw. des Datenverarbeitungslogikzellenfeldes, also Überschreiben bzw. Nachladen und/oder Aktivieren einer neuen Konfiguration, die die eigentliche Verarbeitung der Daten
10 durchführt (in diesem Zusammenhang sei darauf hingewiesen, daß für eine solche Verarbeitungs-konfiguration auch Bezug genommen werden kann auf eine Mehrzahl von Konfigurationen, die z. B. im Wave-Modus und/oder sequenziell nacheinander abzuarbeiten sind) zuzugreifen wie bei einem
15 Vektorregister und dann die dabei erhaltenen Ergebnisse und/oder Zwischenergebnisse wiederum in die internen oder über die XPP wie interne Speicher verwalteten externen Speicher, um dort diese Ergebnisse abzulegen. Die so vektorregisterartig mit Verarbeitungsergebnissen beschriebenen Speichermittel unter XPP-Zugriff sind dann, nach
20 Rekonfigurieren der Verarbeitungs-konfiguration durch Laden der STORE-Konfiguration in geeigneter Weise weggeschrieben, was wiederum datenstromartig geschieht, sei es über den I/O-Port direkt in externe Speicherbereiche und/oder, wie besonders bevorzugt, in Cache-Speicherbereiche, auf welche dann zu
25 einem späteren Zeitpunkt die Sequenziell-CPU und/oder andere Konfigurationen auf der zuvor die Daten erzeugt habenden XPP oder einer anderen entsprechenden Datenverarbeitungseinheit zugreifen können.

30

Eine besonders bevorzugte Variante besteht darin, zumindest für bestimmte Datenverarbeitungsergebnisse und/oder Zwischen-

ergebnisse als Speicher- bzw. Vektorregistermittel, in wel-
 chem bzw. welches die erhaltenen Daten abzulegen sind, nicht
 einen internen Speicher zu benutzen, in welchen Daten über
 eine STORE-Konfiguration in den Cache- oder einen anderen Be-
 5 reich, auf welchen die Sequenziell-CPU oder eine andere
 Datenverarbeitungseinheit zugreifen können, wegzuschreiben
 sind, sondern statt dessen unmittelbar die Ergebnisse wegzu-
 schreiben in entsprechende, insbesondere zugriffsreservierte
 Cachebereiche, die insbesondere Slice-artig organisiert sein
 10 können. Dies kann gegebenenfalls den Nachteil einer größeren
 Latenz haben, insbesondere wenn die Wege zwischen der XPP-
 oder Datenverarbeitungslogikzellenfeldeinheit und dem Cache
 so lang sind, daß die Signallaufzeiten ins Gewicht fallen,
 führt aber dazu, daß gegebenenfalls keine weitere STORE-
 15 Konfiguration benötigt wird. Es sei im übrigen erwähnt, daß
 eine derartige Abspeicherung von Daten in Cache-Bereiche ei-
 nerseits, wie vorstehend beschrieben, dadurch möglich ist,
 daß der Speicher, in welchen geschrieben wird, physikalisch
 nahe beim Cache-Controller liegt und als Cache ausgestaltet
 20 ist, dass aber alternativ und/oder zusätzlich auch die Mög-
 lichkeit besteht, einen Teil eines XPP-Speicherbereiches,
 XPP-internen Speichers oder dergleichen, insbesondere bei RAM
 über PAEs unter die Verwaltung eines oder, nacheinander meh-
 rerer Cache-Speichercontroller zu stellen. Dies hat dann
 25 Vorteile, wenn die Latenz beim Abspeichern der Verarbeitungs-
 ergebnisse, welche innerhalb des
 Datenverarbeitungslogikzellenfeldes bestimmt werden, gering
 gehalten werden soll, während die Latenz beim Zugriff auf den
 dann nur noch als „Quasi-Cache“ dienenden Speicherbereich
 30 durch andere Einheiten nicht oder nicht signifikant ins Ge-
 wicht fällt.

Es sei im übrigen erwähnt, daß auch eine Ausgestaltung derart möglich ist, daß der Cache-Controller einer herkömmlichen Sequenziell-CPU einen Speicherbereich als Cache anspricht, der, ohne dem Datenaustausch mit dem Datenverarbeitungs-

5 logikzellenfeld zu dienen, auf und/oder bei diesem physikalisch liegt. Dies hat den Vorteil, daß dann, wenn Anwendungen auf dem Datenverarbeitungslogikzellenfeld laufen, die einen allenfalls geringen lokalen Speicherbedarf haben, und/oder wenn auch nur wenige weitere Konfigurationen bezogen auf die
 10 zur Verfügung stehenden Speichermengen benötigt werden, diese einer oder mehreren Sequenziell-CPU's als Cache zur Verfügung stehen können. Es sei erwähnt, daß dann der Cache-Controller für die Verwaltung eines Cache-Bereiches mit dynamischem Umfang, d. h. variierender Größe ausgebildet sein kann und
 15 wird. Eine dynamische Cache-Umfangsverwaltung bzw. Cache-Umfangsverwaltungsmittel für die dynamische Cache-Verwaltung wird typisch die Arbeitslast auf der Sequenziell-CPU und/oder dem Datenverarbeitungslogikzellenfeld berücksichtigen. Mit anderen Worten kann beispielsweise analysiert werden, wie
 20 viele NOPs in einer gegebenen Zeiteinheit auf der Sequenziell-CPU vorliegen und/oder wie viele Konfigurationen im XPP-Feld in dafür vorgesehenen Speicherbereichen vorabgelegt sein sollen, um eine schnelle Umkonfiguration, sei es im Wege einer Wellenrekongfiguration oder auf andere Weise. Die hiermit
 25 offenbarte dynamische Cachegröße ist dabei insbesondere bevorzugt laufzeitdynamisch, d. h. der Chacecontroller verwaltet jeweils eine aktuelle Cachegröße, die sich von Takt zu Takt oder Taktgruppe ändern kann. Es sei im übrigen darauf hingewiesen, daß die Zugriffsverwaltung eines XPP-
 30 bzw. Datenverarbeitungslogikzellenfeldes mit Zugriff als interner Speicher wie bei einem Vektorregister und als Cache-artiger Speicher für den externen Zugriff was die Speicherzu-

griffe angeht bereits beschrieben wurde in der DE 196 54 595 und der PCT/DE 97/03013 (PACT03). Die genannten Schriften sind durch Bezugnahme zu Offenbarungszwecken hiermit vollumfänglich eingegliedert.

5

Vorstehend wurde auf Datenverarbeitungslogikzellenfelder Bezug genommen, die insbesondere zur Laufzeit rekonfigurierbar sind. Es wurde diskutiert, dass bei diesen eine Konfigurationsverwaltungseinheit (CT bzw. CM) vorgesehen werden kann.

10 Aus den diversen, zu Offenbarungszwecken unter Bezug genommenen Schutzrechten des Anmelders sowie seinen weiteren Veröffentlichungen ist die Verwaltung von Konfigurationen per se bekannt. Es sei nun explizit darauf hingewiesen, dass derartige Einheiten und deren Wirkungsweise, mit der

15 insbesondere unabhängig von Ankopplungen an Sequenziell-CPUs etc. aktuell noch nicht benötigte Konfigurationen vorladbar sind, auch sehr gut nutzbar sind, um im Multitaskingbetrieb und/oder bei Hyperthreading und/oder Multithreading einen Task- beziehungsweise einen Thread- und/oder Hyperthreadwechsel zu bewirken. Dazu kann ausgenutzt werden, dass während
20 der Laufzeit eines Threads oder Tasks in die Konfigurationspeicher bei einer einzelnen oder einer Gruppe von Zellen des Datenverarbeitungslogikzellenfeldes, also beispielsweise einer PAE eines PAE-Feldes (PA) auch Konfigurationen für

25 unterschiedliche Aufgaben, das heißt Tasks oder Threads beziehungsweise Hyperthreads geladen werden können. Dies führt dann dazu, dass bei einer Blockade eines Tasks oder Threads, etwa wenn auf Daten gewartet werden muss, weil diese noch nicht verfügbar sind, sei es, da sie von einer anderen Einheit
30 noch nicht generiert oder empfangen wurden, beispielsweise auf Grund von Latenzen, sei es, weil eine Resource derzeit noch durch einen anderen Zugriff blockiert

ist, dann Konfigurationen für einen anderen Task oder Thread vorladbar und/oder vorgeladen sind und auf diese gewechselt werden kann, ohne dass der Zeitoverhead für einen Konfigurationswechsel bei der insbesondere schattengeladenen

- 5 Konfiguration abgewartet werden muss. Während es prinzipiell möglich ist, diese Technik auch dann zu verwenden, wenn innerhalb eines Tasks die wahrscheinlichste Weiterführung vorhergesagt wird und eine Vorhersage nicht zutrifft (prediction miss), wird diese Art des Betriebs bei vorhersagefreiem
- 10 Betrieb bevorzugt sein. Bei Verwendung mit einer rein sequentiellen CPU und/oder mehreren rein sequentiellen CPUs wird somit durch die Zuschaltung eines Konfigurationsmanagers eine Hyperthreadingverwaltungshardware realisiert. Verwiesen sei hinsichtlich dessen insbesondere auf PACT10 (DE 198 07 872.2,
- 15 WO 99/44147, WO 99/44120). Dabei kann es als ausreichend erachtet werden, insbesondere dann, wenn nur für eine CPU und/oder einige wenige Sequenziell-CPU's eine Hyperthreadingverwaltung gewünscht ist, auf bestimmte, in den speziell unter Bezug genommenen Schutzrechten beschriebene Teilschal-
- 20 tungen wie den FILMO zu verzichten. Insbesondere wird damit die Verwendung der dort beschriebenen Konfigurationsmanager mit und/oder ohne FILMO für die Hyperthreadingverwaltung für eine und/oder mehrere rein sequenziell arbeitende CPUs mit oder ohne Ankopplung an eine XPP oder ein anderes Datenverar-
- 25 beitungslogikzellenfeld offenbart und hiermit für sich beansprucht. Es wird hierin eine für sich erfinderische Besonderheit gesehen. Es sei im Übrigen erwähnt, dass eine Vielzahl von CPUs realisiert werden kann mit den bekannten Techniken, wie sie insbesondere aus PACT31 (DE 102 12 621.6-
- 30 53, PCT/EP 02/10572) bekannt sind, bei welchen innerhalb eines Arrays eine oder mehrere Sequenziell-CPU's aufgebaut werden unter Ausnutzung eines oder mehrerer Speicherbereiche

insbesondere im Datenverarbeitungslogikzellenfeld für den Aufbau der sequenziellen CPU, insbesondere als Befehls- und/oder Datenregister. Auch sei darauf verwiesen, dass bereits in früheren Anmeldungen wie PACT02, (DE 196 51 075.9-53, WO 98/26356), PACT04 (DE 196 54 846.2-53, WO 98/29952), PACT08, (DE 197 04 728.9, WO 98/35299) offenbart wurde, wie Sequenzer mit Ring- und/oder Wahlfrei-Zugriff-Speichern aufgebaut werden können.

10 Es sei darauf hingewiesen, dass ein Task- beziehungsweise Thread- und/oder Hyperthreadwechsel unter Verwendung der bekannten CT-Technologie derart erfolgen kann und bevorzugt auch erfolgen wird, dass einem per se bekannten, Software-
15 implementierten Betriebssystem-Scheduler oder dergleichen von der CT Performance-Scheiben und/oder Zeitscheiben zugeordnet werden, während welchen bestimmt wird, von welchen Tasks oder Threads nachfolgend welche Teile per se, unterstellt, dass Ressourcen frei sind, abzuarbeiten sind. Dazu sei ein Beispiel wie folgt gegeben: Zunächst soll für einen ersten Task
20 eine Adressfolge generiert werden, gemäß welcher während der Ausführung einer LOAD-Konfiguration Daten aus einem Cache-Speicher, an dem ein Datenverarbeitungslogikzellenfeld in der beschriebenen Weise angekoppelt ist, geladen werden sollen. Sobald diese Daten vorliegen, kann mit der Abarbeitung einer
25 zweiten, der eigentlichen Datenverarbeitungskonfiguration, begonnen werden. Auch diese kann vorgeladen werden, da sicher feststeht, dass diese Konfiguration, sofern keine Interrupts oder dergleichen einen vollständigen Taskwechsel erzwingen, auszuführen ist. In herkömmlichen Prozessoren ist nun das
30 Problem des sogenannten Cache-Miss bekannt, bei dem die Daten zwar angefordert werden, aber nicht im Cache für den Ladezugriff bereit liegen. Tritt ein solcher Fall in einer Kopplung

gemäß der vorliegenden Erfindung auf, kann bevorzugt auf einen anderen Thread, Hyperthread und/oder Task gewechselt werden, der insbesondere zuvor von dem insbesondere softwareimplementierten Betriebssystem-Scheduler und/oder einer

5 anderen hard- und/oder softwareimplementierten, entsprechend wirkenden Einheit für eine nächstmögliche Ausführung bestimmt wurde und demgemäß bevorzugt vorab in einen der verfügbaren Konfigurationsspeicher des Datenverarbeitungslogikzellenfeldes insbesondere im Hintergrund während der Ausführung einer

10 anderen Konfiguration, beispielsweise der LOAD-Konfiguration, welche das Laden jener Daten, auf die nun gewartet wird, bewirkt hat, geladen wurde. Das für die Vorabkonfiguration ungestört von der tatsächlichen Verschaltung der insbesondere grobgranular ausgebildeten Datenverarbeitungslogikzellen des

15 Datenverarbeitungslogikzellenfeldes separate Konfigurationsleitungen von der konfigurierenden Einheit zu den jeweiligen Zellen direkt und/oder über geeignete Bussysteme geführt sein können wie per se im Stand der Technik bekannt, sei hier noch einmal explizit erwähnt, da diese Ausbildung hier besonders

20 bevorzugt ist, um ein ungestörtes Vorabkonfigurieren ohne Störung einer anderen, gerade laufenden Konfiguration zu ermöglichen. Wenn dann die Konfiguration, auf welche während beziehungsweise auf Grund des Task-Thread- und/oder Hyperthreadwechsels gewechselt wurde, abgearbeitet wurde, und

25 zwar, bei bevorzugten nicht teilbaren, ununterbrechbaren und somit quasi atomaren Konfigurationen bis zum Ende abgearbeitet wurde, wird teilweise eine weitere andere Konfiguration wie vorbestimmt durch die entsprechenden Scheduler, insbesondere den betriebssystemartigen Scheduler festgelegt, abgearbeitet

30 und/oder jene Konfiguration, zu welcher zuvor die zugehörige LOAD-Konfiguration ausgeführt wurde. Vor der Ausführung einer Verarbeitungskonfiguration, zu welcher zuvor eine LOAD-

Konfiguration ausgeführt wurde, kann insbesondere abgetestet werden, ob mittlerweile die entsprechenden Daten in das Array eingeströmt sind, also die Latenzzeit, wie sie typisch auftritt, verstrichen ist und/oder die Daten tatsächlich
5 vorliegen.

Mit anderen Worten werden dann Latenzzeiten, wenn sie auftreten, weil z. B. Konfigurationen noch nicht einkonfiguriert sind, Daten noch nicht geladen und/oder Daten noch nicht weg-
10 geschrieben wurden, überbrückt und/oder verdeckt, indem Threads, Hyperthreads und/oder Tasks ausgeführt werden, welche schon vorkonfiguriert sind und welche mit Daten arbeiten, die schon verfügbar sind beziehungsweise die an Ressourcen weggeschrieben werden können, die für das Wegschreiben be-
15 reits zur Verfügung stehen. Auf diese Weise werden Latenzzeiten weitgehend überdeckt und es wird, eine hinreichende Anzahl von per se auszuführenden Threads, Hyperthreads und/oder Tasks unterstellt, eine praktisch 100%-ige Ausnutzung des Datenverarbeitungslogikzellenfeldes erreicht.

20 Mit dem beschriebenen System bezüglich Datenstrom-Fähigkeit bei gleichzeitiger Ankopplung an eine Sequenziell-CPU und/oder bezüglich der Ankopplung eines XPP-Array beziehungsweise Datenverarbeitungslogikzellenfeldes und simultan einer Sequenziell-CPU an eine geeignete Schedulereinheit wie einen
25 Konfigurationsmanager oder dergleichen lassen sich insbesondere ohne weiteres echtzeitfähige Systeme realisieren. Zur Echtzeitfähigkeit muss gewährleistet sein, dass auf eintreffende Daten beziehungsweise Interrupts, die insbesondere das
30 Dateneintreffen signalisieren, innerhalb einer in keinem Fall zu überschreitenden Maximalzeit reagiert werden kann. Dies kann beispielsweise geschehen durch einen Taskwechsel auf ei-

nen Interrupt hin und/oder, beispielsweise bei priorisierten Interrupts, durch Festlegung, dass ein gegebener Interrupt momentan zu ignorieren ist, wobei auch dies innerhalb einer bestimmten Zeit festzulegen ist. Ein Taskwechsel bei derartigen echtzeitfähigen Systemen wird typisch auf drei Arten erfolgen können, nämlich entweder dann, wenn ein Task eine bestimmte Zeit gelaufen ist (Watch-dog-Prinzip), bei Nichtzurverfügungstehen einer Ressource, sei es durch deren Blockade durch anderen Zugriff oder aufgrund von Latenzen beim Zugriff darauf, insbesondere in schreibender und/oder lesender Weise, das heißt bei Latenzen von Datenzugriffen und/oder beim Auftreten von Interrupts.

Mit der vorliegenden Erfindung kann die Echtzeitfähigkeit eines Datenverarbeitungslogikzellenfeldes nunmehr erreicht werden, indem eine oder mehrere von drei möglichen Varianten implementiert wird.

Eine erste Variante dazu besteht darin, dass innerhalb einer von dem Scheduler beziehungsweise der CT ansprechbaren Ressource ein Wechsel zur Abarbeitung beispielsweise eines Interrupts erfolgt. Sofern die Ansprechzeiten auf Interrupts oder andere Anforderungen so groß sind, dass während dieser Zeit eine Konfiguration ohne Unterbrechung noch abgearbeitet werden kann, ist dies unkritisch, zumal während der Abarbeitung der aktuell laufenden Konfiguration auf jener Ressource, die für die Abarbeitung des Interrupts zu wechseln ist, eine Konfiguration zur Interruptabarbeitung vorgeladen werden kann. Die Auswahl der vorabzuladenden Interrupt-bearbeitenden Konfiguration ist z. B. durch die CT durchzuführen. Es ist möglich, die Laufzeit der Konfiguration auf der für die Interruptbearbeitung freizugebenden bzw. zu wechselnden

Ressource zu begrenzen. Verwiesen wird dazu auf
PACT29/PCT(PCT/DE03/000942).

Bei Systemen, die schneller auf Interrupts reagieren müssen,
5 kann es bevorzugt sein, eine einzelne Ressource, also bei-
spielsweise eine separate XPP-Einheit und/oder Teile eines
XPP-Feldes für eine solche Abarbeitung zu reservieren. Wenn
dann ein schnell abzuarbeitender Interrupt auftritt, kann
entweder eine für besonders kritische Interrupts schon vorab
10 vorgeladene Konfiguration abgearbeitet werden oder es wird
sofort mit dem Laden einer Interrupt behandelnden Konfigura-
tion in die reservierte Ressource begonnen. Eine Auswahl der
jeweils für den entsprechenden Interrupt erforderlichen Kon-
figuration ist durch entsprechende Triggerung,
15 Waveabarbeitung usw. möglich.

Es sei im Übrigen erwähnt, dass es mit den schon beschriebe-
nen Methoden ohne weiteres möglich ist, eine instantane
Reaktion auf einen Interrupt zu erhalten, indem über die Ver-
20 wendung von LOAD/STORE-Konfigurationen eine Code-Reentranz
erreicht wird. Hierbei wird nach jeder datenbearbeitenden
Konfiguration oder zu gegebenen Zeiten, beispielsweise alle
fünf oder zehn Konfigurationen eine STORE-Konfiguration aus-
geführt und dann eine LOAD-Konfiguration unter Zugriff auf
25 jene Speicherbereiche ausgeführt, in die zuvor weggeschrieben
wurde. Wenn sichergestellt wird, dass die von der STORE-
Konfigura-tion benutzten Speicherbereiche so lange unberührt
bleiben, bis durch Fortschreiten im Task eine weitere Konfi-
guration sämtliche relevanten Informationen (Zustände, Daten)
30 weggeschrieben hat, ist sichergestellt, dass bei Wiederladen,
also Wiedereintritt in eine zuvor bereits begonnene, aber
nicht zu Ende geführte Konfiguration oder Konfigurationskette

wieder dieselben Bedingungen erhalten werden. Eine solche Zwischenschaltung von LOAD/STORE-Konfigurationen unter simultanem Schutz von noch nicht veralteten STORE-

Speicherbereichen lässt sich automatisch ohne zusätzlichen

5 Programmieraufwand sehr einfach generieren, z. B. von einem Compiler. Dort kann die Ressourcenreservierung gegebenenfalls vorteilhaft sein. Das bei der Ressourcenreservierung und/oder in anderen Fällen auf zumindest eine Menge hochpriorisierter Interrupts durch Vorabladen von bestimmten Konfigurationen
10 reagiert werden kann, sei noch einmal erwähnt.

Eine weitere, besonders bevorzugte Variante der Reaktion auf Interrupts besteht dann, wenn zumindest eine der ansprechbaren Ressourcen eine Sequenziell-CPU ist, darin, auf dieser

15 eine Interrupt-Routine abzuarbeiten, in welcher wiederum Code für das Datenverarbeitungslogikzellenfeld verboten ist. Mit anderen Worten wird eine Interrupt-Routine ausschließlich auf einer Sequenziell-CPU abgearbeitet, ohne dass XPP-Datenverarbeitungsschritte aufgerufen werden. Dies garantiert, dass der

20 Verarbeitungsvorgang auf dem Datenverarbeitungslogikzellenfeld nicht zu unterbrechen ist und es kann dann eine

Weiterabarbeitung auf diesem Datenverarbeitungslogikzellenfeld nach einem Taskswitch erfolgen. Obwohl damit die

25 eigentliche Interrupt-Routine keinen XPP-Code besitzt, kann dennoch dafür gesorgt werden, dass auf einen Interrupt hin zu

einem späteren, nicht mehr echtzeitrelevanten Zeitpunkt mit der XPP auf einen durch einen Interrupt und/oder eine Echtzeitanforderung erfassten Zustand und/oder Daten unter

30 Verwendung des Datenverarbeitungslogikzellenfeldes reagiert werden kann.

Es ist bei der Erfindung möglich, bei einem an eine CPU angekoppelten Datenverarbeitungslogikzellenfeld optimierte Konfigurationen in das Feld zu laden, das insbesondere aus einem Analog/Digital-Mischfeld bestehen kann und Zellen mit

5 frequenzoptimierten Aspektverhältnis besitzt. Beim Laden von Konfigurationen kann es nun sehr vorteilhaft sein, wenn Busse dynamisch aufbaubar sind. Die vorliegende Erfindung offenbart daher zugleich ein Verfahren zum dynamischen Aufbau von Bussen in Feldern miteinander kommunizierender Elemente,

10 insbesondere rekonfigurierbarer Felder wie Prozessoren grobgranularer Felder, das in Verbindung mit den andere Aspekten besonders vorteilhaft ist, aber zugleich auch für sich genommen erfinderisch-

15 Es ist bereits bekannt, grobgranulare Felder rekonfigurierbarer Elemente vorzusehen, bei denen zwischen den rekonfigurierbaren Elementen Bussysteme verlaufen. In bekannten Anwendungen werden die Bussysteme, die die Verbindungen für die Kommunikation der einzelnen Elemente untereinander

20 vorsehen, durch eine zentrale Einheit konfiguriert. Die Art und Weise, wie die Busverbindung aufgebaut werden soll, kann dabei im Vorfeld bestimmt werden, etwa zu einer Compilezeit. Denkbar ist auch eine Festlegung zur Laufzeit, bei der durch einen Scheduler oder dergleichen für verschiedene, aktuell zu

25 ladende Konfigurationen ein Busaufbau vorgenommen wird, bzw. ein Routing. Auf die Anmeldung 102 36 272.8 wird hierzu insbesondere hingewiesen, da diese Anmeldung bereits zeigt, wie aus unterschiedlichen Konfigurationen zur Ausführung ein und desselben Programmes zu einer Laufzeit eine Auswahl erfolgen

30 kann.

Es sind auch schon Bussysteme für rekonfigurierbare Prozessoren bekannt, bei denen ein dynamischer Busaufbau erfolgen kann. Erwähnt sei, daß es insbesondere möglich ist, Bussysteme zu mischen, nämlich die bekannten, sogenannt „globalen“ dynamisch aufbaubaren Busse und nicht dynamisch aufbaubare Busse. Dies gilt auch für die nachfolgend offenbarten Bussysteme und -verfahren, das heißt die beschriebenen Bussysteme und Verbindungsaufbauverfahren müssen nicht die einzigen, in einem Feld zu verbindender Elemente vorzusehenden Bussysteme bzw. -verfahren sein.

Es ist auch möglich, und dies gilt auch für Zwecke der vorliegenden Erfindung, neben grobgranularen Einheiten mit insbesondere feingranularer Steuerlogik, wie feingranularen Triggernetzwerken usw. eine sogenannte Makrogranularität vorzusehen, bei der eine Vielzahl grobgranularer Elemente mit herkömmlichen Bussystemen usw. zusammengefasst werden und bei denen dann mehrere solcher zusammengefasster, grobgranularer Elemente, zwischen denen bereits Bussysteme in konfigurierbarer oder fester Weise vorgesehen sein können, wiederum über Bussysteme kommunizierende Teile einer übergeordneten Einheit bilden. Hierarchische Strukturen für derartige Anordnungen sind beispielsweise aus DE 199 26 538.0 od. PCT WO 00/77652 bekannt.

Es ist oftmals wünschenswert, Busse dynamisch aufzubauen, besonders dann, wenn ein Prozessor für Multitasking, Multithreading, Hyperthreading usw. verwendet werden soll, und/oder insbesondere wenn sehr große Felder aus zum Beispiel 65.536 PAEs oder mehr aufgebaut werden sollen.

In einem solchen Fall ist es wünschenswert, für eine automatische, selbsttätige, d.h. selbsterstellende dynamische Verbindung von Start- und Zielfeldern innerhalb eines solchen Feldes sorgen zu können. Als Start- und/oder Zielelemente können auch neben den aus der herkömmlichen XPP-Technologie bekannten PAEs Elemente vorgesehen werden wie IO-Ports, feldinterne Speicher, Speicher-Ios, FPGAs, sequentielle CPUs, Sequencer, FSM (Finit-State-Machines) reine Lesespeicher, reine Schreibspeicher, NILdevices, etc.

10

Die vorliegende Erfindung schlägt somit in einem weiteren Grundgedanken ein Verfahren zum dynamischen Aufbau einer Verbindung zwischen einem Sender und einem Empfänger über einen einer Vielzahl möglicher Wege, der von Station zu Station fortschreitet, vor, bei welchem vorgesehen ist, dass ausgehend von einer den Busaufbau bewirkenden Einheit (Sender und/oder Empfänger) eine Anfrage an nächstliegende Stationen gesandt wird, welche für den Busaufbau bereit stehen, diesen Stationen eine Kennziffer, hier gleichbedeutend, Kenngröße, zugeordnet wird, ausgehend von wenigstens einer Vielzahl, bevorzugt jeder freien Station, der eine Kennziffer zugeordnet wurde, eine Anfrage an nächstliegende Stationen nach der Verfügbarkeit der Stationen für einen Busaufbau gesandt wird, den verfügbaren Stationen eine weitere Kennziffer zugeordnet wird, und dies fortgesetzt wird, bis das gewünschte Ende des Busses erreicht ist.

30

Ein weiterer wesentlicher Aspekt der vorliegenden Erfindung nutzt somit die Erkenntnis aus, dass Busse problemfrei dadurch aufgebaut werden können, dass an nächstliegende Übertragungsstationen auf dem Weg eines möglichen Busses Anfragen gesandt werden, ob diese Stationen für den Busaufbau

bereit stehen und dann, ausgehend von bereitstehenden Stationen, in einem weiteren Schritt diese nächstliegenden Stationen angesprochen werden, wobei durch die Kennzifferzuordnung eine Ansprechreihenfolge festgehalten wird, um anhand derselben eine Rückverfolgung des Busaufbaus zu ermöglichen. Es ist dabei zwar möglich, dass nicht von jeder angesprochenen und als frei erkannten Station beim Busaufbau weitergeschritten wird, weil etwa eine Auswertung in der Station eines gewünschten Zielpunktes zeigt, dass der Busaufbau sich weit in eine falsche Richtung bewegt hat, bevorzugt wird aber von jeder freien Station, der eine Kennziffer zugeordnet wurde, versucht, den Bus weiter aufzubauen, indem die Nachbarstationen der zuerst angesprochenen Station gleichfalls angesprochen werden.

Hintergrund hierfür ist, dass es Situationen geben kann, etwa bei einem bereits nahezu vollen Array, in welches weitere Konfigurationen hineingesetzt werden sollen, wo es erforderlich ist, einen Busaufbau über größere Umwege zuzulassen, um, sollte ein Busaufbau überhaupt möglich sein, diesen sicher zu ermöglichen.

In einer bevorzugten Variante wird regelmäßig jeder Station, die angesprochen wurde, eine Kennziffer zugewiesen. Dies ist vorteilhaft, um festzustellen, dass die Station bereits angesprochen wurde und somit nicht mehr bei Ansprechen aus einer anderen Richtung vermeintlich zur Verfügung steht. Dies verhindert, dass eine Signalausbreitung noch stattfindet, nachdem die Nachbarstationen bereits wieder als nicht benötigt freigegeben wurden.

In einer besonders bevorzugten Variante wird die Kenngröße von Station zu Station so verändert, dass der Weg, der beim Busaufbau gewählt wird, nachvollziehbar ist, insbesondere im Wege eines Backtracing. Dieses Backtracing kann durch Inkrementierung oder Dekrementierung eines am Ziel erreichten Wertes erfolgen, insbesondere mit jeweils fester Schrittweite. Bei Vorsehen einer festen Schrittweite kann insbesondere auch ein zyklisches Zählen erfolgen, d. h. ein Zählen in einem zyklischen Zahlenraum, bei dem nach Überschreiten des höchst möglichen Wertes immer wieder mit einem kleineren Wert begonnen wird (z. B. 1, 2, 3, 4; 1, 2, 3, 4; 1, 2, 3, 4; oder 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5;). Dabei ist dann zur Charakterisierung der Station, um ein einwandfreies Rückverfolgen des Weges zu gewährleisten, ein zyklisches Durchzählen von mindestens drei verschiedenen Zählwerten bevorzugt.

Das beschriebene Verfahren wird, sofern überhaupt ein Busaufbau zwischen Sender und Empfänger möglich ist, diesen aufzubauenden Bus identifizieren. Beim Busaufbau werden aber womöglich eine Vielzahl nicht benötigter Stationen angesprochen und es ist daher bevorzugt, diese wieder freizugeben, und zwar nach Aufbau des Busses bzw. mit der Signalisierung zwischen Sender und Empfänger, dass ein Busweg aufgebaut wurde. Dazu kann ausgehend von der letzten, den Aufbau vollendenden Station, typisch als dem Signalempfänger, wenn der Busaufbau ausgehend vom Sender hin zum Empfänger erfolgt, rückwärts unter Abschreitung der Kennwerte die davorliegende Station angesprochen werden und sichergestellt werden, dass die von dieser Station angesprochenen, anderen und somit nicht auf dem Bus(rück)weg liegenden Stationen für eine Fremdnutzung freigegeben werden. Dabei wird von jeder ange-

5 gesprochen und für die weitere Benutzung in anderen Buswegen
freigegebenen Stationen wiederum fortgeschritten zu allen an-
deren, nicht benötigten Stationen, die zuvor angesprochen
wurden. Auf diese Weise ist sichergestellt, dass alle Statio-
nen, die zuvor für den Busaufbau angesprochen wurden, nunmehr
wieder zur Verfügung stehen.

Es sei erwähnt, dass neben diesem Verfahren zur rückschrei-
tenden Freigabe eines aufgebauten Busweges andere
10 Möglichkeiten existieren, nach Aufbau eines Busweges zwischen
Sender und Empfänger nicht benötigte Stationen freizugeben.
So kann etwa längs aller Stationen, die für den Busweg benö-
tigt werden, ein Signal gesandt werden, welches den
Busstationen mitteilt, dass sie am Busweg beteiligt sind. Ei-
15 ne solche Information kann wiederum rückwärts gesandt werden
im Wege des Back-tracing, etwa indem die Kennziffern ausge-
wertet werden, die den Stationen während der Aufbauphase
zugeordnet wurden. Danach kann eine globale Freigabe erfol-
gen, etwa indem ausgehend von der Anfangsstation oder einer
20 zentralen Kontrollinstanz sämtliche, momentan nicht an beste-
henden Bussen verwendeten Stationen zurückgesetzt bzw. für
den Aufbau eines Busweges freigegeben werden.

Es sei darauf hingewiesen, dass etwa auch unter bestimmten
25 Bedingungen, wie nach einem festen Zeitablauf, eine Busfrei-
gabe erfolgen kann. Diese Art der Freigabe kann allerdings
verhindern, dass Busse, die per se aufbaubar wären, aufgebaut
werden können; so ist bei sehr großen Prozessorfeldern denk-
bar, dass die Wege extrem lang werden, weil um verschiedene,
30 im Laufe des Betriebes dynamisch in das Feld hineinkonfigu-
rierte Zellgruppenanordnungen mäanderförmig ein Weg herum
bzw. durch solche Konfigurationen gelegt werden muss, was bei

großen Feldern sehr lange dauern kann. Sorge ist daher bevorzugt dafür zu tragen, dass eine hinreichende Zeit für den Aufbau eines Busses verbleibt.

- 5 Es sei erwähnt, dass es prinzipiell möglich ist, insbesondere in sehr großen Feldern, simultan mehrere Buswege bzw. Busverbindungen zwischen unterschiedlichen Stationen und unterschiedlichen Empfängern aufzubauen. Dies kann jedoch dazu führen, dass sich zwei aufzubauende Busverbindungen
- 10 wechselseitig im Fortschreiten blockieren, so dass keiner von den beiden Bussen erfolgreich eine Verbindung aufbaut. Es kann mit anderen Worten zu einem Deadlock kommen. Dass derartige Deadlock-Situationen auch bei simultanem Aufbau mehrerer Busse verhindert werden können, sei erwähnt. So ist vorstell-
- 15 bar, Bussen eine Priorität zuzuordnen und damit zu gewährleisten, dass dann, wenn ein aufzubauender Bus hoher Priorität auf einen gleichfalls noch nicht aufgebauten Bus geringerer Priorität trifft, die Stationen des Busses mit geringerer Priorität besetzt werden, d. h. dass die vorherige
- 20 Reservierung für einen aufzubauenden Bus geringerer Priorität ignoriert wird. Dabei wird die tatsächliche Implementierung derartiger Verbindungsaufbauten abhängig davon zu gestalten sein, wie die für die Busaufbauprotokollumsetzung erforderliche Logik in eine Halbleiterarchitektur umzusetzen ist, d. h.
- 25 welcher Aufbau im Einzelfall erforderlich ist; wie der Busaufbau und erforderlichenfalls der Versuch eines Neuaufbaus nach Scheitern eines ersten Versuchs auszusehen hat, ob eine Priorisierung erfolgen soll und kann, wobei vorstellbar ist, eine Priorisierung eines aufzubauenden Busses etwa nach Bedeutung des Makros, das in das Feld konfiguriert wird, der
- 30 Wartezeit seit dem Versuch eines ersten Aufbaus usw., zu bestimmen.

Prinzipiell wäre es möglich, nach dem Erreichen des Ziels ausgehend vom Start, also typisch nach Erreichen des Empfängers ausgehend vom den Busaufbau typisch bewirkenden Sender, lediglich ein Signal auszusenden, das dem Sender anzeigt, dass überhaupt ein Bus aufbaubar ist, damit dieser mit dem Senden beginnen mag. In einem solchen Fall könnte ein abzusendendes Datenpaket einfach wie eine Stationsaufbauanfrage an alle Nachbarstationen gesandt werden. Es müsste dann aber mit jedem Datenpaket sichergestellt werden, dass am Empfänger erkannt werden kann, von wo, das heißt von welcher Station ein abgesandtes Datenpaket zuerst empfangen wird, es muss sichergestellt werden, dass ein bestimmtes Datenpaket nur einmal empfangen wird, auch wenn es über andere, verschlungene Pfade später noch einmal zum Empfänger gelangt. Bevorzugt wird aber in jedem Fall sein, dass die anderen Stationen freigegeben werden, etwa durch Back-tracing nach Erreichen der Zielstation. Dieses Busbeteiligungssignal, das rückwärts gesandt wird, kann ausgehend von den Zahlenwerten, die den Nachbarstationen zugeordnet werden, erfolgen. Im übrigen sei darauf hingewiesen, dass es gleichfalls möglich ist, dass sich die Station lediglich merkt, aus welcher Richtung aus sie angesprochen wurde. In einem solchen Fall kann ohne Vergleich an den Nachbarstationen, welche Kennzifferwerte diese besitzen, eine Rückverfolgung sehr schnell geschehen und es kann überdies dann, wenn in der Station festgehalten wird, welche Nachbarstationen beim Busaufbau noch angesprochen wurden, gewährleistet werden, dass auch beim Back-tracing die nicht am aufgebauten Bus beteiligten Stationen freigegeben werden.

Die einer Station beim Ansprechen zuzuordnende Kennziffer kann daher auch eine Kennziffer sein, die die Richtung angibt, von welcher aus die Station angesprochen wurde. Bei vier nächsten anzusprechenden Nachbarn reichen dafür beispielsweise zwei Bit. Werden zusätzlich noch die Stationen gespeichert, die angesprochen worden sind, während der Bus aufgebaut wurde, sind weitere vier Bit erforderlich bei einer Vier-nächste-Nachbarn-Architektur. Es kann ein weiteres Bit hinzugefügt werden, um zu kennzeichnen, ob die Station überhaupt schon angesprochen wurde oder vom Busaufbau des aktuell aufzubauenden Busses bislang unberührt geblieben war. Kommen Priorisierungen und dergleichen dazu, sind weitere Zustände festzuhalten. Es sei darauf hingewiesen, dass dies feingranular erfolgen kann, und zwar insbesondere auch dann, wenn das Prozessorfeld für sich grobgranular aufgebaut ist.

Es sei weiter darauf hingewiesen, dass verschiedene Möglichkeiten existieren, nach dem erfolgreichen Aufbau eines ersten Busses zwischen einem ersten Sender und einem ersten Empfänger den Aufbau eines zweiten Busses zwischen z. B. einem zweiten Sender und einem zweiten Empfänger zu ermöglichen; dabei kann auch einer der Sender und/oder einer der Empfänger identisch sein. Das Ansprechen von zwei Empfängern aus ein und demselben Sender kann sinnvoll sein, wenn etwa ein Berechnungsergebnis als Eingang für zwei verschiedene Zweige eines Programmes, die in unterschiedliche Bereiche hineinkonfiguriert sind, benötigt wird. Ein Ansprechen eines einzelnen Empfängers aus mehreren Sendern kann erwünscht sein, wenn dort etwa zwei Operanden, die aus unterschiedlichen Konfigurationsbereichen erhalten werden sollen, zu verknüpfen sind, und ein Ansprechen von einem Empfänger über ein und denselben Sender kann geboten sein, wenn Operanden, die zu unterschied-

lichen Zeiten erhalten bzw. bestimmt wurden, an ein und demselben Empfänger miteinander verknüpft werden sollen, etwa in der Form $a_n \times a_{n-1}$. Es kann dann über Register im Bus sichergestellt werden, dass eine solche Verknüpfung nach Aufbau von

5 zwei Bussystemen möglich wäre, auch wenn dies typisch schon aus Gründen des Energieverbrauches im Bussystem weniger bevorzugt wäre als eine lokale Zwischenspeicherung von Operanden und dergleichen. Der Aufbau des weiteren oder

10 nächst aufzubauenden Busses kann etwa dadurch erfolgen, dass mit dem Stationsfreigabesignal nach vorläufiger Reservierung einer Station ein Signal mitgesandt wird, welches angibt, welchem aufgebauten Bus die Station gehörte, wobei dieser Bus etwa wiederum durch ein Priorisierungssignal gekennzeichnet werden kann. Wenn eine freigebende Station dann einer Station

15 benachbart liegt, die gerne selbst einen Bus aufbauen würde, der eine geringfügig niederere Priorisierung hat, kann dies dort festgestellt werden und es kann dann von dieser Station ausgehend der nächste Busaufbau bewirkt werden. Alternativ kann, bei einer globalen Freigabe aller momentan nicht an ei-

20 nem Busaufbau benötigten Stationen und/oder danach ein globales Signal abgesandt werden, etwa von einer zentralen Kontrollinstanz, welches dem Feld mitteilt, welche Busverbindung als nächste aufzubauen ist bzw. welche Priorität die nächste aufzubauende Busverbindung haben soll. Statt eines

25 globalen Broadcast einer derartigen Busaufbauverwaltungsinformation kann auch insbesondere zentral und/oder an mehreren Stellen dezentral, etwa bei hierarchisch angeordneten Prozessorfeldern, bei denen innerhalb eines bestimmten Bereiches ein Busaufbau gewünscht wird, lokal und/oder regional eine

30 Signalisierung an eine einen Busaufbau anfordernde Station wie einen Sender, der seinen Empfänger erreichen muss, erfolgen.

Welche Art der Stationsfreigabe und/oder der Mitteilung, dass ein weiterer Bus aufgebaut werden kann, tatsächlich implementiert wird, wird insbesondere davon abhängig sein, wie

5 schnell die diesbezügliche Information über das Array propagiert werden kann und/oder welche Busaufbauhäufigkeit über der Zeit erwartet wird. So kann dann, wenn eine Analyse zeigt, dass die auf einem Feld typisch benötigten und simultan abzuarbeitenden Konfigurationen selten einen Busaufbau
10 erfordern, der zudem langsam erfolgen kann, eine prozessorarchitektonisch einfache Implementierung gewählt werden, die mit wenigen Logikelementen auskommt, um die entsprechende Steuerung zu gewährleisten, während bei sehr häufigem und auch schnell benötigtem Busaufbau eine komplexere Implementierung
15 sinnvoll sein kann.

In einer besonders bevorzugten Variante ist es möglich, unter mehreren per se bezüglich der Buslänge und/oder des Umfangs der Stationen entlang des Busses gleichwertiger Bussysteme
20 einen Bus auszuwählen anhand verschiedener objektiver Beurteilungskriterien. Während es prinzipiell möglich ist, in einem solchen Fall eine zufällige Wahl zu treffen, können, je nach Anforderung und tatsächlicher Auslegung, unterschiedliche Kriterien herangezogen werden. Es kann etwa bei

25 Architekturen, die unterschiedliche Busverbindungen in Horizontal- und Vertikalrichtung aufweisen, etwa wenn die Busverbindungen in Vertikalrichtung zusätzlich Register umfassen, durch welche die Daten zu schleusen sind, während längs der Vertikalrichtung registerfrei und somit Daten mit
30 geringeren Energieverlusten weiterleitende Busverbindungen existieren (ein Beispiel für eine solche Architektur ist die XPP 128 des Anmelders), vorgesehen sein, dass beim Busaufbau

vermerkt wird, wie viele Schritte horizontal und vertikal gegangen wurden. Diese Information kann in einer Station abgelegt werden oder an einem Header, der mit dem Busaufbauanforderungssignal mitübertragen wird; es erfolgt dann zur
 5 Busauswahl eine Auswertung derartiger Informationen. Alternativ kann etwa an jeder Station abgefragt werden, wie viele Busse in der Nähe der Station bereits existieren, um etwa zu ermöglichen, dass eine näherungsweise gleichmäßige Busverbindungs-
 10 dichte über das Array hinweg erhalten wird. Dieses Vorgehen ist einerseits vorteilhaft, weil der Datentransport längs der Busse auf Grund der erforderlichen Umladung der Busleitungskapazitäten, den in Bussen zu integrierenden Treibern usw. einen erhöhten Energieverbrauch ergibt, weshalb eine Vergleichmäßigung der Busverteilungsdichte über das Pro-
 15 zessorfeld zu einer Vergleichmäßigung der thermischen Belastung führt. Insofern kann durch die Vergleichmäßigung insgesamt womöglich die Taktrate bei gleicher Kühlung erhöht werden, was im Bereich mobiler Prozessoren für Laptops, Handys und dergleichen vorteilhaft ist. Andererseits ist eine
 20 Busverbindungs-dichtevergleichmäßigung auch vorteilhaft für die Auslastungserhöhung und Ressourcenschonung.

Schutz wird auch beansprucht für ein multidimensionales Feld rekonfigurierbarer Elemente, in welchem Bussysteme für den
 25 dynamischen Selbstaufbau in einer der vorbeschriebenen Weisen und/oder in einer Weise, wie sie aus dem nachfolgenden ersichtlich ist, vorgesehen sind. Es sei darauf hingewiesen, dass die Bezeichnung „multidimensionales Feld rekonfigurierbarer Elemente“ sich auf grobgranulare rekonfigurierbare
 30 Elemente mit Elementen wie ALUs, erweiterten ALUs, RAMPAs usw. beziehen kann, wie sie vorerwähnt wurden, und dass im Sinne der Erfindung eine Multidimensionalität nicht nur durch

die räumliche Übereinanderanordnung und Nebeneinanderanordnung von rekonfigurierbaren Elementen erhalten werden kann, sondern auch durch eine bestimmte Art und Weise der Verbindung. So sind in linear angeordneten Feldern den Elementen in der Mitte zwei nächste Nachbarn zugeordnet, in zweidimensionalen Feldern wie bei einer Kachelung typisch vier nächste Nachbarn, und in einer dreidimensionalen Anordnung typisch sechs nächste Nachbarn, wie dies aus der Stapelung von Würfeln und dergleichen erkannt werden kann. Die Verwendbarkeit etwa drei- oder sechseckiger Zellen sei beispielhaft erwähnt. Es ist aber auch möglich, zusätzlich Busverbindungen vorzusehen, die diagonal verlaufen, übernächste Nachbarn verbinden, größere Strecken vorsehen usw. Wird eine solche Busstruktur implementiert, ergibt sich eine Multidimensionalität mit einem Dimensionsmaß größer 1, wobei dieses Dimensionsmaß auch von einer ganzen Zahl verschieden sein kann. Eine solche Anordnung wird in jedem Fall als multidimensionales Feld der vorliegenden Erfindung betrachtet.

Die vorliegende Erfindung wird im Folgenden nur beispielsweise an Hand der Zeichnung beschrieben. In dieser zeigt:

Fig. E1 ein multidimensionales Feld miteinander kommunizierender rekonfigurierbarer Elemente, die für den Busaufbau ausgebildet sind, vor dem Beginn des Busaufbaus;

Fig. E2 das Feld von Fig. 1 nach dem ersten Busaufbauschritt;

Fig. E3 das Feld von Fig. 1 nach dem zweiten Busaufbauschritt;

Fig. E4 das Feld von Fig. 1, nachdem das Empfängerfeld erreicht wurde, mit unterschiedlichen, möglichen Busverbindungen;

Fig. E5 die Anordnung mit dem ausgewählten Bus.

5

Nach Fig. 1 umfasst ein allgemein mit 1 bezeichnetes Feld 1 eine Vielzahl rekonfigurierbarer Zellen, die miteinander über selbstaufbauende Busse kommunizieren können.

- 10 Jede Zelle 1a, 1b, 1c usw., die am Busaufbau zu beteiligen ist, weist interne Logikelemente auf, die es ermöglichen, Informationen darüber zu speichern, ob die Zelle derzeit von einem Bus bereits verwendet wird (mit X gekennzeichnete Zellen in Feld 1), ob die Zelle bereits als mögliche Buszelle
- 15 bei einem aktuellen Busaufbau angesprochen wurde, und, wenn ja, in wie vielen vertikalen und horizontalen Schritten der Busaufbau bis zur Zelle erfolgte, wie viele Schritte insgesamt beim Busaufbau zurückgelegt wurden oder ob die Zelle noch vollständig frei ist und noch nicht angesprochen wurde.
- 20 Um die Anzahl der horizontal bzw. vertikal von einem Bus bereits abgeschrittenen Zellen auf dem Weg zwischen einer möglichen Senderzelle S und einer möglichen Empfängerzelle e abzulegen, sind in jeder Zelle zwei Speicherbereiche vorgesehen, die in den Figuren mit H und V bezeichnet sind. Weiter
- 25 ist ein Speicherbereich für die Gesamtzahl der durchlaufenen Schritte ablegbar, wie durch die grossen Zahlen 1 - 12 in den Figuren 1 - 5 dargestellt. Dabei ist die gewählte Maximalzahl 12 nur beispielhaft, da dies im gewählten Beispiel geringer Komplexität die erforderliche Schrittzahl zum Erreichen des
- 30 Empfängers, ausgehend vom gewählten Sender, ist. Die Zellen sind weiter dazu ausgebildet, dann, wenn sie ein Busaufbauanforderungssignal erhalten und frei sind, an einem

aufzubauenden Bus teilzunehmen, und zugleich an Nachbarstationen in einem nachfolgenden Schritt eine Anfrage zu senden, ob diese Nachbarstationen gleichfalls für den Busaufbau frei sind. Dazu weisen sie Signal-Sende- und -Empfangsverbindungs-
 5 beschaltungen für die jeweils nächsten Nachbarn auf. Die einzelne Zelle ist weiter so ausgebildet, dass zusammen mit dem Busaufbauanforderungssignal Information bezüglich der insgesamt schon zurückgelegten Schrittweite und der Anzahl der horizontalen und vertikalen Teilschritte (H und V) an die
 10 angesprochenen Stationen übermittelt werden kann.

Der Busaufbau geschieht im vorliegenden Fall wie folgt:
 Zunächst wird das dynamisch konfigurierbare Array derart betrieben, dass alle Busse aufgebaut seien. Es mögen dann
 15 bestimmte Konfigurationen enden und es sei erforderlich, eine neue Konfiguration fragmentiert in freie Bereiche des Arrays hineinzukonfigurieren, weil eine hinreichende Anzahl funktionell geeigneter Zellen derzeit nicht zur Verfügung steht. Es sei dabei weiter ein Fall gegeben, bei welchem alle Felder
 20 bis auf die mit X bezeichneten für den Busaufbau zur Verfügung stehen.

Nun wird von jenen Zellen, die miteinander zu kommunizieren haben, um ein in das Array hineinzukonfigurierendes Makro
 25 ausführen zu können, eine Sende- und eine Empfangszelle festgelegt. Dies kann durch die Konfiguration bzw. den Scheduler oder ähnliches geschehen. Diese sind in Fig. 1 mit S bezeichnet. Nun sendet die den Busaufbau bewirkende Senderzelle S ein erstes Busaufbauanforderungssignal an ihre unmittelbaren
 30 Nachbarn, das heisst jene Zellen, die an deren Zellenkanten angrenzen, im dargestellten Beispiel also an vier Zellen. Diese Zellen stellen fest, dass sie frei sind, dass sie die

ersten Stationen sind, die Busaufbauanforderungssignale erhalten, und dass sie jeweils einen Schritt horizontal oder vertikal respektive von der Sendezelle entfernt sind. Es wird nun in den Nachbarzellen entsprechend in den H- und V-

5 Speicherbereich eine 0 bzw. 1 eingespeichert, und es wird eine 1 in den Schrittweitenspeicher der angefragten Zelle gespeichert.

Im zweiten Schritt spricht jede zuvor angesprochene freie
 10 Zelle wiederum ihre eigenen Nachbarzellen an und fragt bei diesen an, ob sie für den Busaufbau zur Verfügung stehen. Dies führt dazu, dass eine Reihe weiterer Zellen danach erkannt haben, dass sie für den Busaufbau benötigt werden und die zweiten Zellen im Verlauf eines möglicherweise aufgebauten Busses darstellen. Weiter werden entsprechende Vermerke
 15 über die horizontale bzw. vertikale Schrittweite in entsprechenden Speicherbereichen abgelegt. Die bereits mit X gekennzeichneten Zellen ignorieren hingegen das Busaufbauanforderungssignal, wie das in der 4. Zelle von links, 2. Zeile
 20 von unten der Fall ist.

Nachdem die ersten Zellen ihre Nachbarzellen angesprochen haben, ist klar, dass sie in weiteren Busaufbausritten schweigen können; ein Busanforderungssignal wird so nur un-
 25 mittelbar im Schritt nach jenem ausgesandt, welcher die das Busaufbauanforderungssignal aussendende Zelle reserviert hat. Dies verhindert zwar, dass Zellen, die erst während des Busaufbaus freigegeben werden, später noch reservierbar sind, spart aber, da nicht immer wieder von allen bereits reser-
 30 vierten Zellen Busaufbauanforderungssignale ausgesandt werden müssen, was Treiberleistung erfordert, Energie und ist somit etwa für mobile Anwendungen bevorzugt, wo der erhaltene Vor-

teil überwiegt gegenüber Lösungen, bei denen später freiwerdende Zellen auch noch in einen entstehenden Bus miteinbezogen werden können; hier wäre allerdings besonders etwa Sorge dafür zu tragen, dass stets in jenen Nachbarzellen der Bus-aufbau als relevant eingestuft wird, die die geringsten Schrittweiten längs des Busses erfordern. Im nächsten Busaufbauschritt sprechen nun die zweiten Zellen ihre jeweiligen Nachbarzellen wiederum an, wobei, da die Zellen 1 für den Busaufbau schon reserviert sind, die Zellen 2 nicht mehr zurücklaufen können, sondern nur nach vorne, weg vom Sender. Dies setzt sich fort, bis der Empfänger schließlich erreicht ist, vergleiche Fig. 4.

Es sind nun im Beispiel am Empfänger zwei Zellen gleichzeitig angekommen, die beide die gleiche Schrittweite 12 tragen und es ist, wie durch die verschieden gestrichelten Linien ersichtlich, möglich, über diese Zellen rückwärtsschreitend unterschiedliche Buswege aufzubauen. Es wäre hier zwar prinzipiell eine zufällige Auswahl möglich, es wird aber, wie bevorzugt, zunächst bei jedem Abschreiten in Rückwärtsrichtung versucht, die V-Werte maximal zu halten. Dies führt zu dem durchgehend gezeichneten Bus in Fig. 5. Sobald der Busaufbau durch Rückwärtsschreiten bestätigt wurde, können alle Zellen, die nicht daran beteiligt sind, abgewiesen und wieder freigegeben werden. Dazu wird ein globales Busfreigabesignal ausgestrahlt, das anzeigt, dass alle derzeit nicht an einem aufgebauten Bus beteiligten Zellen sich zurücksetzen können.

Es sei erwähnt, dass die Art und Weise des Busaufbaus durch dynamische Selbstorganisation unter Verwendung geeigneter, dem Durchschnittsfachmann aus der Offenbarung ersichtlichen Hardwareschaltungen in der Zelle festlegbar ist.

Patentansprüche

1. Verfahren zur Auswahl eines aus einer Vielzahl von Wegen
5 zur Erzielung eines Datenverarbeitungsergebnisses bei der
Datenverarbeitung unter zumindest möglicher Verwendung
multidimensionaler Felder konfigurierbarer Datenhandhabungselemente, dadurch gekennzeichnet, daß den
Datenhandhabungselementen konfigurationsabhängig lei-
10 stungsaufnahmebezogene kennzeichnende Größen zugeordnet
werden und eine Wegauswahl unter Zuordnungsbewertung erfolgt.
2. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß einer aus einer Vielzahl
15 unterschiedlicher Algorithmen ausgewählt wird.
3. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet,
20 daß eine aus einer Vielzahl unterschiedlicher Konfigurationen ausgewählt wird.
4. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet,
25 daß über die Zuordnung einer gegebenen Datenverarbeitungsaufgabe und/oder -teilaufgabe an ein multidimensionales
Feld konfigurierbarer Datenhandhabungselemente und/oder
einer anderen Datenverarbeitungsarchitektur, insbesondere
an ein ASIC, und/oder an eine rein sequenziell arbeitende
30 CPU bei der Wegauswahl entschieden wird.

5. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß ein zweidimensionales Feld konfigurierbarer Datenhandhabungselemente betrachtet wird.

5

6. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß eine Auswahl aus einem zur Laufzeit konfigurierbaren, insbesondere störungsfrei partiell rekonfigurierbaren Prozessorfeld vorgenommen wird.

10

7. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß als konfigurierbare Datenhandhabungselemente zumindest einige, bevorzugt alle der Elemente Busse, Register, ALUs, RAMs und/oder andere grobgranulare Datenhandhabungselemente betrachtet werden.

15

8. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß der kennzeichnende Wert unabhängig von der tatsächlichen aktuell zugeordneten Aufgabe nur davon abhängig gewählt wird, ob das jeweilige Datenhandhabungselement überhaupt verwendet wird oder nicht.

20

25

9. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß jeder Konfiguration eine eigene Kennzahl bei der Verwendung zugeordnet wird.

30

10. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß eine strom-, arbeits- und/oder lei-

stungsbezogene Größe als verbrauchs kennzeichnende Größe zugeordnet wird.

11. Verfahren nach dem vorhergehenden Anspruch, dadurch
5 gekennzeichnet, daß eine Auswahl unter Berücksichtigung
eines Summenwertes verbrauchs kennzeichnender Größen erfolgt.

12. Verfahren nach dem vorhergehenden unabhängigen An-
10 spruch oder einem Unteranspruch hierzu , dadurch
gekennzeichnet, daß eine Auswahl unter Berücksichtigung
weiterer Größen erfolgt, insbesondere einer erforderlichen
Ausführungszeit und/oder erforderlicher Ressourcen, einer
aktuellen Prozessorauslastung und/oder einer aktuell ge-
15 wünschten und/oder möglichen Leistungsaufnahme.

13. Verfahren nach dem vorhergehenden Anspruch, dadurch
gekennzeichnet, daß eine Auswahl vor der eigentlichen Da-
tenverarbeitung erfolgt.

20

14. Verfahren nach dem vorhergehenden unabhängigen An-
spruch oder einem Unteranspruch hierzu , dadurch
gekennzeichnet, daß die Kennzeichnungsgrößen bei der Simu-
lation einer Datenverarbeitung bereitgestellt und/oder
25 ausgewählt werden.

25

15. Verfahren nach dem vorhergehenden unabhängigen An-
spruch oder einem Unteranspruch hierzu , dadurch
gekennzeichnet, daß eine Auswahl zur Laufzeit erfolgt.

30

16. Verfahren zum Betrieb eines multidimensionalen Feldes
rekonfigurierbarer Elemente,

worin Anordnungen von

Gruppen zusammen datenhandhabender Elemente in vorbestimmter Weise während der Laufzeit zur Abarbeitung vorgegebener Aufgaben in das Feld hinein konfiguriert werden,

dadurch gekennzeichnet, daß

für zumindest eine abzuarbeitende Aufgabe eine Mehrzahl von solchen Elementgruppenanordnungen im multidimensionalen Feld bestimmt wird, die zur Abarbeitung der

vorgegebenen Aufgabe geeignet sind,

für die Abarbeitung der vorgegebenen Aufgabe eine dann besonders geeignete Elementgruppenanordnung aus der Mehrzahl ausgewählt wird und

die ausgewählte in das Feld hinein konfiguriert wird.

17. Verfahren nach dem vorhergehenden Anspruch dadurch gekennzeichnet, daß die Elementgruppenanordnung anhand ihrer Geometrie im Vergleich mit der Geometrie im Feld bereits für die Umkonfiguration verfügbarer und/oder vermutlich bald verfügbar werdender Elemente ausgewählt wird.

18. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, worin Elementgruppenanordnungen betreffende Konfigurationsdaten aus einem Speicher mit nicht vernachlässigbaren Zugriffszeiten eingelesen werden und/oder mit nicht vernachlässigbaren Generierungszeiten generiert werden, dadurch gekennzeichnet, daß zumindest für einige Konfigurationen für die Auswahl zunächst nur ein gegenüber dem Konfigurationsdatensatz aller Konfigurationsdaten in der Größe reduzierter Kenndatensatz, insbesondere ein Satz von die Geometrie betreffenden Kenndaten, in eine

Elementgruppenanordnungsauswahlstufe eingelesen werden, eine Auswahl aufgrund des Kenndatensatzes getroffen wird und dann im Ansprechen auf die Auswahl die Konfigurationsdaten aus dem Speicher gelesen oder generiert werden.

5

19. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , worin bei der Umkonfiguration des Feldes zur Durchführung einer Mehrzahl zumindest teilweise simultan auszuführender Aufgaben, von denen für wenigstens zwei Gruppen zusammen Daten handhabender Elemente in vorbestimmter Weise während der Laufzeit zur Abarbeitung vorgegebener Aufgaben in das Feld hinein konfiguriert werden, eine Mehrzahl von solchen Elementgruppenanordnungen, die zur Abarbeitung der vorgegebenen Aufgabe geeignet sind, vorbestimmt werden, die zu einer gegebenen Zeit und/oder einem gegebenen Ereignis für die zumindest teilweise simultane Hineinkonfiguration verfügbarer Ressourcen bestimmt werden und jene Elementgruppenanordnungen aus der vorbestimmten Anzahl ausgewählt werden, mit denen eine simultane Abarbeitung besonders effizient möglich ist.

20. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß eine erste Elementgruppenanordnung in das Feld hineinkonfiguriert wird, mit dieser Elementgruppenanordnung begonnen wird, die Aufgabe abzuarbeiten, bis ein vorgegebenes Ereignis auftritt und danach unter zumindest partieller Rekonfiguration mit der Aufgabenabarbeitung in einer weiteren Elementgruppenanordnung fortgefahren wird.

21. Verfahren nach dem vorhergehenden unabhängigen Anspruch
oder einem Unteranspruch hierzu , worin die Elementgrup-
penanordnungen sich hinsichtlich der
Abarbeitungsgeschwindigkeit unterscheiden und die Auswahl
5 im Ansprechen auf die gegebenenfalls unter Berücksichti-
gung weiterer Aspekte maximal erzielbare
Abarbeitungsgeschwindigkeit erfolgt.
22. Multidimensionales Prozessorfeld aufweisend
10 eine Vielzahl benachbart angeordneter Datenverarbeitungs-
zellen mit
Eingängen, die Daten von Vernetzungswegen erhalten,
einer Operanden-Verknüpfungseinheit, die diese entspre-
chend der jeweiligen Funktion ihrer Operanden-
15 Verknüpfungseinheit verknüpfen und
Ausgängen, um die Daten verknüpft auf Vernetzungswege
aufzugeben,
dadurch gekennzeichnet, daß
die Datenverarbeitungszellen ein Aspektverhältnis aufwei-
20 sen, das wenigstens 2:1, bevorzugt 2:1 beträgt.
23. Multidimensionales Prozessorfeld nach dem vorhergehenden
Anspruch, dadurch gekennzeichnet, daß die Datenverarbei-
tungszellen als grobgranular konfigurierbare Zellen
25 ausgebildet sind.
24. Multidimensionales Prozessorfeld nach dem vorhergehenden
unabhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß Datenverarbeitungszellen in
30 Reihen und Spalten angeordnet sind.

25. Prozessorfeld nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß zumindest bei einem Teil der Datenverarbeitungszellen Dateneingänge vorgesehen sind, um von einer oberen Reihe Daten zu erhalten und Datenausgänge, um an eine untere Reihe Daten auszugeben.

26. Prozessorfeld nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß die Datenverarbeitungseinheiten EALUs, ALU und/oder registerflankierte Zellen sind.

10

27. Verfahren zur Konfigurierung eines Prozessorfeldes nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, worin Zellen für die Konfiguration ausgewählt und in Funktion und Vernetzung bestimmt werden, wobei eine Vernetzung derart bestimmt wird, daß Daten von Zelle zu Zelle zumindest weitgehend verzögerungsfrei übertragbar sind, dadurch gekennzeichnet, daß als benachbarte Zellen, zwischen denen Daten binnen eines Taktes oder einer geringen Taktzahl übertragbar sind, auch solche berücksichtigt werden, die nicht unmittelbar nebeneinander liegen, sondern in der Breite durch eine Strecke getrennt sind, die geringer ist als die Länge der Zelle.

25 28. Multidimensionales Prozessorfeld aufweisend eine Vielzahl benachbart angeordneter Datenverarbeitungszellen mit

Eingängen, die Daten von Vernetzungswegen erhalten, einer Operanden-Verknüpfungseinheit, die diese entsprechend der jeweiligen Funktion ihrer Operanden-Verknüpfungseinheit verknüpfen und Ausgängen, um die Daten verknüpft auf Vernetzungswege

30

aufzugeben,

dadurch gekennzeichnet, daß
die Datenverarbeitungszellen ein Aspektverhältnis aufwei-
sen, das wenigstens 2:1, bevorzugt 2:1 beträgt.

5

29. Multidimensionales Prozessorfeld nach dem vorhergehenden
Anspruch, dadurch gekennzeichnet, daß die Datenverarbei-
tungszellen als grobgranular konfigurierbare Zellen
ausgebildet sind.

10

30. Multidimensionales Prozessorfeld nach dem vorhergehenden
unabhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß Datenverarbeitungszellen in
Reihen und Spalten angeordnet sind.

15

31. Prozessorfeld nach dem vorhergehenden Anspruch, dadurch
gekennzeichnet, daß zumindest bei einem Teil der Datenver-
arbeitungszellen Dateneingänge vorgesehen sind, um von
einer oberen Reihe Daten zu erhalten und Datenausgänge, um
an eine untere Reihe Daten auszugeben.

20

32. Prozessorfeld nach dem vorhergehenden Anspruch, dadurch
gekennzeichnet, daß die Datenverarbeitungseinheiten EALUs,
ALU und/oder registerflankierte Zellen sind.

25

33. Verfahren zur Konfigurierung eines Prozessorfeldes nach
dem vorhergehenden unabhängigen Anspruch oder einem Un-
teranspruch hierzu , worin Zellen für die Konfiguration
ausgewählt und in Funktion und Vernetzung bestimmt werden,
wobei eine Vernetzung derart bestimmt wird, daß Daten von
Zelle zu Zelle zumindest weitgehend verzögerungsfrei über-
tragbar sind, dadurch gekennzeichnet, daß als benachbarte

30

Zellen, zwischen denen Daten binnen eines Taktes oder einer geringen Taktzahl übertragbar sind, auch solche berücksichtigt werden, die nicht unmittelbar nebeneinander liegen, sondern in der Breite durch eine Strecke getrennt sind, die geringer ist als die Länge der Zelle.

34. Datenverarbeitungsanordnung mit einem multidimensionalen Feld

10 in Funktion und/oder Vernetzung konfigurierbarer Zellelemente und

diesen zugeordneten Konfigurationsvorhaltemitteln zum lokalen Konfigurations-Vorhalten, dadurch gekennzeichnet, daß
15 die Konfigurationsvorhaltemittel dazu ausgebildet sind, zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorzuhalten.

35. Datenverarbeitungsanordnung mit einem multidimensionalen Feld

20 in Funktion und/oder Vernetzung konfigurierbarer Zellelemente und

diesen zugeordneten Konfigurationsvorhaltemitteln zum lokalen Konfigurations-Vorhalten, dadurch gekennzeichnet, daß
25 die Konfigurationsvorhaltemittel dazu ausgebildet sind, zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorzuhalten.

30 36. Datenverarbeitungsanordnung nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, dadurch gekennzeichnet, daß die Funktion grobgranular kon-

figurierbar ist.

37. Datenverarbeitungsanordnung nach dem vorhergehenden un-
abhängigen Anspruch oder einem Unteranspruch hierzu ,
5 dadurch gekennzeichnet, daß die Vernetzung grobgranular
konfigurierbar ist.

38. Datenverarbeitungsanordnung nach dem vorhergehenden un-
abhängigen Anspruch oder einem Unteranspruch hierzu ,
10 dadurch gekennzeichnet, daß als Zellelemente zumindest ei-
nes von ALUs, EALUs, RAM-Zellen, I/O-Zellen, Logiblöcken
vorgesehen sind.

39. Datenverarbeitungsanordnung nach dem vorhergehenden un-
15 abhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß jedem Zellelement ein eigenes
Konfigurationsvorhaltemittel zugeordnet ist.

40. Datenverarbeitungsanordnung nach dem vorhergehenden un-
20 abhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß die Konfigurationsvorhaltemit-
tel dazu ausgebildet sind, eine Vielzahl von
Konfigurationen vorzuhalten.

25 41. Datenverarbeitungsanordnung, dadurch gekennzeichnet, daß
mehrere fest vorgegebene nichtflüchtige Konfigurationen im
Konfigurationsvorhaltemittel vorgegeben sind.

42. Datenverarbeitungsanordnung nach dem vorhergehenden un-
30 abhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß die Anordnung dazu ausgebildet
ist, eine wechselnde einer Vielzahl von vorgehaltenen Kon-

figurationen zu verwenden, insbesondere im Wege der Wave-Rekonfiguration oder des lokalen Sequencing.

43. Datenverarbeitungsanordnung nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß bei eingien Zellen im Betrieb mit veränderlichen Konfigurationen versehbare Konfigurationsvorhaltemittel vorgesehen sind.

44. Datenverarbeitungsanordnung nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu ,
dadurch gekennzeichnet, daß als Konfigurationsvorhaltemittel zumindest eines aus ROM, EPROM, EEPROM, Flash-Speicher, Fuse-, Antifuse-programmierbare Speichermittel und/oder in insbesondere in oberen Lagen einer Siliziumstruktur fest vorgesehene Speichermittel gewählt sind.

45. Verfahren zur Herstellung einer dedizierten Datenverarbeitungsanordnung, dadurch gekennzeichnet, daß ein multidimensionales Feld mit in Funktion und/oder Vernetzung konfigurierbaren Zellelemente und diesen zugeordneten Konfigurationsvorhaltemitteln zum lokalen Konfigurationsvorhalten vorgegeben wird, bestimmt wird, welche Konfigurationen in diesen vorzuhalten sind, und dann nichtflüchtige Konfigurationsvorhaltemittel so vorgesehen werden, daß sie zumindest einen Teil der vorgehaltenen Konfigurationen nichtflüchtig vorhalten.

46. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß von einem zur laufzeitrekfigurierbaren multidimensionalen Feld ausgegangen wird.

47. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, daß zunächst von einem zur
laufzeitrekonfigurierbaren multidimensionalen Feld mit re-
konfigurationsbeschaltung ausgegangen wird und dann für
5 rekonfiguration nichtbenötigte Felder weggelassen werden.

48. Rekonfigurierbare Signalverarbeitungsvorrichtung mit ei-
ner Vielzahl von miteinander, insbesondere konfigurierbar
verknüpfbaren Signalverarbeitungsschaltkreisen, die in
10 ihrer Funktion veränderlich sind, wobei diese einen Aus-
gang aufweisen, über welchen eine Umkonfigu-rierung
angefordert und/oder eine Akzeptierung einer Umkonfigura-
tion vorgesehen werden kann, dadurch gekennzeichnet, dass
zumindest ein Teil der verknüpfbaren Signalverarbeitungs-
15 schaltkreise analoge Signalverarbeitungsschaltkreise
sind, wobei weiter eine Rekonfigurationseinheit zur Vor-
gabe von Konfigurationen für die das Analogsignal
verarbeitenden Schaltkreise vorgesehen ist.

20 49. Rekonfigurierbare Signalverarbeitungsvorrichtung nach dem
vorhergehenden Anspruch, dadurch gekennzeichnet, dass ein
Teil der Signalverarbeitungsschaltkreise digitale Schalt-
kreise sind.

25 50. Datenverarbeitungsvorrichtung mit einem Datenverar-
beitungslogikzellenfeld und zumindest einer Sequenziell-
CPU, dadurch gekennzeichnet, dass eine Ankopplung der Se-
quenziell-CPU und des Datenverarbeitungslogikzellenfeldes
zum Datenaustausch in insbesondere blockweiser Form durch
30 zu einem Cache-Speicher führende Leitungen möglich ist.

50. Verfahren zum dynamischen Aufbau

einer Verbindung zwischen einem Sender und einem Empfänger

über einen einer Vielzahl möglicher Wege, der von Station zu Station fortschreitet,

dadurch gekennzeichnet, dass

ausgehend von einer den Busaufbau wirkenden Einheit (Sender und/oder Empfänger) eine Anfrage an nächstliegende Stationen gesandt wird, welche für den Busaufbau bereit stehen,

diesen Stationen eine Kennziffer zugeordnet wird, ausgehend von wenigstens einer Vielzahl, bevorzugt jeder freien Station, der eine Kennziffer zugeordnet wurde, eine Anfrage an nächstliegende Stationen nach der Verfügbarkeit der Stationen für einen Busaufbau gesandt wird,

den verfügbaren Stationen eine weitere Kennziffer zugeordnet wird, und dies fortgesetzt wird, bis das gewünschte Ende des Busses erreicht ist.

51. Verfahren nach dem vorhergehenden Anspruch, worin von

Station zu Station, die jeweils für ein Fortschreiten des Busaufbaus verfügbar ist, unabhängig davon, ob mit dieser Station das Ziel erreicht werden wird oder nicht, eine veränderte Kennziffer zugewiesen wird.

52. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass die Veränderung von Station zu Station

nachvollziehbar gewählt wird, wobei insbesondere eine Incrementierung oder Decrementierung eines Zählwertes um

eine feste Größe erfolgt, insbesondere mit Schrittweite gegebenenfalls durch zyklisches Zählen, d.h. Zählen in einem endlichen, zyklischen Zahlenraum.

5 53. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass zyklisch bis mindestens drei verschiedene Zählwerte zur Charakterisierung der Station durchgezählt werden.

10 54. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, daß nach Aufbau eines Bus-Weges zwischen Sender und Empfänger nicht benötigte Stationen wieder freigegeben werden.

15 55. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu , dadurch gekennzeichnet, dass ein Bus als aufgebaut betrachtet wird, sobald von einer Station aus das Ziel erreicht worden ist.

20 56. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass nach dem Aufbau eines ersten Busses der Aufbau weiterer Busverbindungen unterbunden wird.

25 57. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass den an einem aufgebauten Bus beteiligten Stationen das Erreichen des Busaufbaues mitgeteilt wird.

30 58. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass den an einem aufgebauten Bus beteiligten Stationen das Erreichen des Busaufbaues mit-

geteilt wird,

indem ausgehend vom Ziel in Rückwärtsrichtung den am Bus beteiligten Stationen ein Busbeteiligungssignal zugesandt wird, insbesondere durch Übertragen von Signalen entlang des Busses.

59. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass die Identifikation der am aufgebauten Bus beteiligten Stationen erfolgt, indem ausgehend vom Ziel die den erreichbaren Stationen gegebenenfalls zugeordneten Zahlenwerte verglichen werden und jene Stationen ausgewählt werden, die einen buszugehörigkeitsindikativen Zahlenwert besitzen.

60. Verfahren nach dem vorhergehenden Anspruch, worin ausgehend von einer Startposition die Zahlenwerte incrementiert werden, während beim Busaufbau von Station zu Station fortgeschritten wird und worin nach Erreichen des Ziels jene Stationen als zum Bus zugehörig identifiziert werden, die jeweils die kleinsten Zahlenwerte erreichbarer Stationen besitzen.

61. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass eine Freigabe nicht für eine Busverbindung benötigter Stationen durch Markierung jener Stationen, die an einem aufgebauten, benötigten Bus teilhaben, und Freigeben aller nicht dergestalt markierten Stationen, insbesondere durch das Senden eines globalen Freigabesignals erfolgt.

62. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, dadurch gekennzeichnet,

dass die Freigabe nicht benötigter Stationen wellenartig durch Zurücklaufen von einer Ziel- zu einer Starteinheit erfolgt und/oder durch Rückwärtslaufen vom Ziel zum Start unter Freigabe nicht benötigter Stationen durch Senden eines Freigabesignals durch jeweils jene Station, die bei der freizugebenden Station zuvor angefragt hatte, ob sie für den Busaufbau zur Verfügung stehe.

63. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, worin simultan eine Vielzahl von Busverbindungen aufgebaut wird, dadurch gekennzeichnet, dass zu jeder für den Aufbau einer Busverbindung angesprochenen Station ein Richtungswert zugeordnet wird, insbesondere in der Station abgespeichert wird, der angibt, aus welcher Richtung oder von welcher ansprechenden Station ein Busaufbauanforderungssignal erhalten wurde.

64. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass die Busverifikation durch Auswertung der Signale erfolgt, die anzeigen, von welcher ansprechenden Station aus ein Busaufbauanforderungssignal gesandt wurde und worin der Busaufbau zu dieser anfordernden Station hin erfolgt.

65. Verfahren nach dem vorhergehenden unabhängigen Anspruch oder einem Unteranspruch hierzu, dadurch gekennzeichnet, dass bei Vorliegen mehrerer, eine gleiche Anzahl von Stationen benötigender Busse, die potenziell aufbaubar sind, ein Bus anhand eines dem Bus zuweisbaren und/oder zugewiesenen Beurteilungskriterium ausgewählt wird.

5 66. Verfahren nach dem vorhergehenden Anspruch, dadurch gekennzeichnet, dass als Busbeurteilungskriterium die Anzahl horizontal und/oder vertikal durchlaufender Stationen und/oder die Anzahl registerbehafteter Stationen und/oder die Verbindungsdichte existierender Busverbindungen längs der Stationen und/oder die Größe verbleibender freier Felder, die nicht durch den Bus verteilt werden, herangezogen wird.

10 67. Multidimensionales Feld rekonfigurierbarer Elemente, wobei zumindest zwischen einigen der konfigurierbaren Elementen dynamisch aufbaubare Busse vorgesehen sind, dadurch gekennzeichnet, dass im Bussystem Einheiten vorgesehen sind, die dazu ausgebildet sind, direkt oder
15 über Fernbusse indirekt benachbarte Gruppen in Bussegmenten anzusprechen und/oder eine Busstationsnummer zu speichern und/oder eine Beförderungssignalrichtung zu speichern und/oder einen Abbau von temporär gehaltenen Verbindungen zu womöglich für einen Busaufbau benötigten
20 Stationen nach Feststellung des Nichtbenötigens durch Freigabe zu ermöglichen.

25 68. Datenverarbeitungsvorrichtung mit einem Datenverarbeitungslogikzellenfeld und zumindest einer Sequenziell-CPU, dadurch gekennzeichnet, dass eine Ankopplung der Sequenziell-CPU und des Datenverarbeitungslogikzellenfeldes zum Datenaustausch in insbesondere blockweiser Form durch zu einem Cache-Speicher führende Leitungen möglich ist.